

VŠB - Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky

DIPLOMOVÁ PRÁCE

2011

Jakub Wagner

VŠB - Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra informatiky

**Popis procesů a návrh systému
pro Českou speleologickou společnost**
**Process Modeling and System Design
for Czech Speleological Society**

Zadání

Poděkování

Rád bych na tomto místě poděkoval všem, kteří mě vedli při tvorbě této práce. Především pak panu Štěpánovi Kuchařovi, za jeho trpělivost, přívětivé chování a ochotu, protože bez toho by tahle práce jistě nevznikla. Dále bych chtěl poděkovat všem členům České speleologické společnosti, za příjemnou spolupráci a výdrž při analýze problémových částí.

Prohlášení

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

V Ostravě dne 28. května 2011

Wagner Jakub

Abstrakt

Cílem této práce je popsat a analyzovat základní procesy České speleologické společnosti dle jejich priorit a navrhnout, jak tyto procesy urychlit, zjednodušit či jinak zdokonalit. Hlavním vylepšením je vývoj informačního systému specializovaného pro tuto společnost, který zajistí jednodušší komunikaci členů společnosti a kvalitnější přehlednost informací o osobách a aktivitách. Funkce jednotlivých uskupení v rámci společnosti jsou zabaleny do oddělených modulů, což umožňuje přidávání nových funkcí, aniž by se narušily původní. Celý projekt je vytvořen na principech procesu softwarového vývoje RUP a jednotlivé vizuální prezentace nebo modelování jsou zpracovány univerzálním grafickým jazykem UML.

Klíčová slova

speleologie, základní organizace, komise, informační systém, analýza procesů

Abstract

The aim of this thesis is to describe and analyze the basic processes of the Czech Speleological Company according to their priorities and suggest how these processes can be sped up, simplified and improved. The main improvement is the development of an information system specialized for the company which provides easier communication among its members and provides better transparency of information about persons and activities. The function of each group within the company is packaged into separate modules that allow adding new functionality without modifying the original functions. The entire project is built on the principles of the RUP software development process and individual visual presentations or models are handled using universal graphical language UML.

Keywords

speleology, elementary organization, commission, information system, process analysis

Seznam použitých symbolů a zkratek

API	Application Programming Interface
CVS	Concurrent Version System
ČSS	Česká speleologická společnost
FTP	File Transfer Protocol
J2SE	Java 2 Platform, Standard Edition
JDBC	Java DataBase Connectivity
NT	New Technology
OMG	Object Management Group
RUP	Rational Unified Process
SQL	Structured Query Language
SZS	Speleologická záchranná služba
ToPo	Topografický program
UML	Unified Modeling Language
URL	Unified Resource Locator
XP	Experience
ZO	Základní organizace

Obsah

1	Úvod	9
2	Unifikovaný modelovací jazyk	10
2.1	Základní prvky jazyka UML	10
2.1.1	Předměty	10
2.1.2	Vztahy	11
2.1.3	Diagramy	11
3	Rational Unified Process	17
3.1	Fáze vývoje	17
3.2	Základní principy	18
3.2.1	Iterativní vývoj	18
3.2.2	Správa požadavků	19
3.2.3	Komponentová architektura	19
3.2.4	Vizuální modelování	20
3.2.5	Průběžné ověřování kvality	20
3.2.6	Řízení změn	20
3.3	Disciplíny RUP	20
3.3.1	Byznys modelování	21
3.3.2	Specifikace požadavků	21
3.3.3	Analýza a návrh	21
3.3.4	Implementace	22
3.3.5	Testování	22
3.3.6	Nasazení	22
3.3.7	Řízení změn a konfigurací	22
3.3.8	Projektové řízení	22
3.3.9	Správa prostředí	22
4	Česká speleologická společnost	23
4.1	Struktura ČSS	23
4.1.1	Orgány sdružení	24
4.1.2	Organizační jednotky sdružení	25
4.1.3	Pracovní komise	25
4.2	Základní názvosloví	26
5	Systém pro ČSS	27
5.1	Nastavení priorit	27
5.2	Nový systém – 1. iterace	28
5.2.1	Byznys modelování – 1. iterace	28
5.2.2	Specifikace požadavků – 1. iterace	34
5.2.3	Analýza – 1. iterace	37
5.2.4	Návrh – 1. iterace	38
5.2.5	Implementace – 1. iterace	43
5.2.6	Testování – 1. iterace	43
5.2.7	Ostatní disciplíny – 1. iterace	44
5.3	Nový systém – 2. iterace	44
5.3.1	Byznys modelování – 2. iterace	44

5.3.2	Specifikace požadavků – 2. iterace	49
5.3.3	Analýza a návrh – 2. iterace.....	51
5.3.4	Ostatní disciplíny – 2. iterace.....	52
6	Závěr.....	53
7	Reference	54
8	Přílohy	55

Seznam obrázků

Obrázek 2-1	Diagram případů užití	12
Obrázek 2-2	Diagram aktivit	13
Obrázek 2-3	Sekvenční diagram.....	14
Obrázek 2-4	Třídní diagram	15
Obrázek 2-5	Diagram nasazení.....	16
Obrázek 3-1	Fáze vývoje.....	17
Obrázek 3-2	Iterativní vývoj	18
Obrázek 3-3	Pohled 4 + 1	19
Obrázek 4-1	Obecná struktura ČSS	23
Obrázek 5-1	Organizační struktura ZO.....	28
Obrázek 5-2	Základní procesy předsedy ZO	28
Obrázek 5-3	Základní procesy pokladníka ZO.....	29
Obrázek 5-4	Základní procesy člena ZO.....	29
Obrázek 5-5	Diagram aktivit: Evidence členů ZO.....	30
Obrázek 5-6	Diagram aktivit: Explorace a prolongace jeskyní	31
Obrázek 5-7	Diagram aktivit: Mapování.....	32
Obrázek 5-8	Diagram aktivit: Sledování chiropterologické fauny	32
Obrázek 5-9	Struktura rolí.....	34
Obrázek 5-10	Diagram případů užití - Funkční specifikace požadavků	35
Obrázek 5-11	Sekvenční diagram: Přihlášení a odhlášení	35
Obrázek 5-12	Sekvenční diagram: Editování člena v evidenci	36
Obrázek 5-13	Třídní diagram: Základní struktura objektů.....	38
Obrázek 5-14	Diagram komponent: Pohled implementace.....	40
Obrázek 5-15	Diagram nasazení: Pohled nasazení	40
Obrázek 5-16	Třídní diagram: Statický model systému.....	42
Obrázek 5-17	Organizační struktura komise pro práci ve výškách a nad volnou hloubkou	45
Obrázek 5-18	Základní procesy předsedy komise	45
Obrázek 5-19	Základní procesy člena komise.....	45
Obrázek 5-20	Diagram aktivit: Řízení komise	46
Obrázek 5-21	Diagram aktivit: Evidence vyškolených pracovníků a instruktorů	47
Obrázek 5-22	Diagram aktivit: Školení instruktorů.....	48
Obrázek 5-23	Struktura rolí - 2. iterace.....	49
Obrázek 5-24	Diagram případů užití: Funkční specifikace požadavků - 2. iterace	50

1 Úvod

Česká speleologická společnost [1] je nezisková organizace se složitou strukturou orgánů vykonávajících spoustu procesů, které se navzájem proplétávají. Je zde zavedení jistý systém, jak tyhle procesy řídit a vykonávat, ale povětšinou se jedná o metodiku, která byla zavedena několik desítek let zpět a nyní jsme schopni ji zdokonalit.

Výsledným cílem však není jen zdokumentování a vylepšení postupů v ČSS, ale také analyzovat potřeby jednotlivých orgánů a jejich členů, a tím vytvořit základy pro systém, který by zjednodušil a urychlil jejich komunikaci či práci s administrativou. Pro názorné zpracování této analýzy je nutné zvolit vhodný modelovací jazyk, který by nám umožnil popis veškerých potřeb a požadavků na systém. Unifikovaný modelovací jazyk umožňuje specifikovat právě tuhle problematiku a jeho základy jsou vysvětleny v kapitole **2 Unifikovaný modelovací jazyk** [2].

Jak už bylo řečeno, k analýze procesů budeme používat metody iterativního postupu vývoje, což je také základem moderního rámce procesu vývoje softwaru RUP. RUP je plně podporován jazykem UML, což zajistí kompatibilní průběh práce. Více o metodice RUP je popsáno v kapitole **3 Rational Unified Process** [3].

Pro vytvoření návrhů na vylepšení si musíme nejprve nastínit strukturu ČSS, jelikož se jedná o rozsáhlou společnost, této tématice se věnuje kapitola **4 Česká speleologická společnost**. Zde je také umístěn seznam pojmů a aktivit, které jsou specifické pro tuto organizaci a běžnému čtenáři by mohly být neznámé.

Hlavní náplní práce je však analýza interních procesů této společnosti a vytvoření systému, který se bude iterativně rozvíjet a bude na sebe nabalovat podporu dalších požadavků členů ČSS. Každá iterace obsahuje jak byznys modelování, tak specifikace požadavků, analýzu a návrh. Součástí je také implementovaný prototyp. Kompletní popis roztřízený podle iterací je umístěn v kapitole **5 Systém pro ČSS**.

2 Unifikovaný modelovací jazyk

Unified Modeling Language zkráceně jazyk UML je univerzální grafický jazyk sloužící k vizualizaci, specifikaci, navrhování a dokumentaci systémů. V dnešní podobě je otevřeným rozšiřitelným průmyslovým standardem pro vizuální modelování, který je schválen sdružením OMG (Object Management Group [4]). [2]

UML podporuje objektově orientovaný přístup k analýze, návrhu a popisu programových systémů, ale neobsahuje způsob, jak se má používat a neobsahuje metodiku, jak analyzovat, specifikovat či navrhovat programové systémy. Jazyk UML modeluje systém nejen jako kolekci objektů, ale také definuje vztahy a komunikace mezi nimi. To je právě jeho velkou výhodou, ale je možné jej vhodně použít i k modelování jiných systémů, jak to můžeme vidět i v této práci, kde bylo jazyku UML využito jak při softwarové analýze samotného systému, tak při modelování interních procesů společnosti.

UML dokáže modelovat jak statické, tak dynamické aspekty, které se navzájem doplňují. Statický aspekt poukazuje na objekty, které se budou v systému vyskytovat, popisuje jejich typ, souvislosti mezi těmito typy a také instance jednotlivých typů. Dynamický aspekt popisuje stejné objekty, ale v závislosti na čase, jak budou mezi sebou komunikovat a navzájem spolupracovat, aby dosáhly požadovaných výsledků.

2.1 Základní prvky jazyka UML

Jelikož je UML vizuálním jazykem, dbá hlavně na přehlednost zobrazení různých pohledů na systém. Proto existují jen tři druhy základních prvků, které se užívají: předměty (things), vztahy (relationships) a diagramy (diagrams) [5].








2.1.1 Předměty

Předměty jsou základním prvkem jazyka UML, které obsahuje každý model. Nemusí být však jen objekty, proto se dělí na strukturní předměty (structural things), předměty chování (behavioral things), předměty seskupení (grouping things) a popisné předměty (annotation things) [5].

- **Strukturní předměty** jsou podstatná jména v modelech, jako například: třída, případy užití, spolupráce, aktivity, interface, uzly apod.
- **Předměty chování** jsou aktivity, děje, nebo také slovesa modelů, jako například: interakce, stav apod.
- **Předměty seskupení** slouží k zabalení sémanticky souvisejících prvků modelu do balíků.
- **Popisné předměty** slouží k přidávání poznámek a dodatečných podmínek k jednotlivým předmětům v modelu.

2.1.2 Vztahy

Vztahy existují jen ve spojení mezi dvěma nebo více předměty a popisují, jaký má jeden předmět vliv na druhý nebo jak mezi sebou významově souvisí [5].

Vztah	Syntaxe UML	Popis
Asociace (Association)		Popis množiny možných spojení mezi objekty. Spojení obsažené v jedné asociaci mají stejnou strukturu a význam. K asociacím se může připojit informace o násobnosti spojení s jednotlivými předměty, nebo také směr těchto spojení.
Agregace (Aggregation)		Slabý vztah části k celku, kdy daná část může existovat samostatně a nemůže být součástí více celků. Jedná se o speciální typ asociace.
Kompozice (Composition)		Silný vztah části k celku, kdy daná část nemůže existovat samostatně a nemůže být součástí více než jednoho celku. Jedná se o speciální typ asociace.
Závislost (Dependency)		Jedná se o vztah mezi poskytovatelem a odběratelem služby. Kdykoliv dojde ke změně u poskytovatele, může to mít vliv na význam závislého prvku. Směr závislosti vede od odběratele k poskytovateli.
Zobecnění (Generalization)		Jeden prvek je specializací druhého, tedy druhý jej minimálně obsahuje. Zobecnění zde směřuje od specializovaného prvku k obecnému.
Realizace (Realization)		Jedná se o vztah třídy a rozhraní, kdy třída musí realizovat dohody dle popisu rozhraní. Realizace směřuje od třídy k realizovanému rozhraní.
Spojení (Link)		Spojení mezi objekty, slouží taky jako ukazatel na poznámky popisných předmětů.

2.1.3 Diagramy

Diagramy jsou také základním stavebním prvkem jazyka UML. Jsou to vizuální prvky, které se skládají z předmětů a vztahů modelu a existují v několika formách, aby poskytly zobrazení systému z nejrůznějších úhlů pohledu. V každém typu se vyskytují předměty v jiných souvislostech, aby jej bylo možné detailně upřesnit [5].

Jak už bylo řečeno, jazyk UML dokáže modelovat dynamické i statické aspekty a každý z těchto typů k tomu využívá určité diagramy, které také dělíme do dvou skupin.

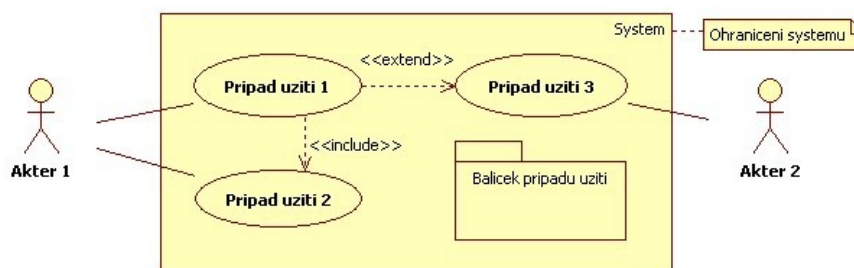
- **Diagramy struktury** zahrnují všechna zobrazení jazyka UML ze skupiny statických a řadíme mezi ně diagramy tříd, diagramy složené struktury, diagramy komponent, diagramy nasazení, objektové diagramy a diagramy balíčků.

- **Diagramy chování** naopak zahrnují všechna zobrazení jazyka UML ze skupiny dynamických a patří mezi ně diagramy aktivit, sekvenční diagramy, diagramy komunikace, diagramy přehledu interakce, diagramy časování, diagramy případů užití a stavové diagramy.

Není vždy nutné využít všechny z vypsanych diagramů, některé slouží jen na speciální případy, když informace podané jiným diagramem nejsou dostačující nebo zavádějící. Dále bych rád rozebral sémantiku a syntaxi těch diagramů, které jsem při své práci využil.

Diagram případů užití – Use Case Diagram

Tento diagram zobrazuje funkce nebo možnosti systému, které poskytuje daným uživatelům. Tyto možnosti se nazývají scénáře a reprezentují právě případ užití. Uživatel nebo přesněji aktér a případy užití jsou pak propojeny, což znamená, že aktér dané případy užití vyvolává za účelem dosažení určitých cílů. Na obrázku 2-1 jsou zobrazeny všechny typy předmětů a vztahů tohoto diagramu.



Obrázek 2-1 Diagram případů užití

Aktér znázorňuje aktivní entitu, může se jednat o člověka, ale i to může být signál jiného systému, který spolupracuje s daným případem užití.

Případ užití popisuje reakci ve formě funkce nebo popisu chování systému na požadavek aktérů. Aktéři se pomocí případů užití snaží dosáhnout svého cíle a systém interaktivně reaguje tak, aby chránil zájmy všech aktérů.

Balíček případů užití slouží k seskupení případů užití s podobnou sémantikou. Může obsahovat buď případy užití samotné, nebo je strukturován do dalších balíčků.

Vztah <<include>> spojuje případy užití a znázorňuje, že když se provádí jeden případ užití, tak se využívá i ten, který k němu patří (include).

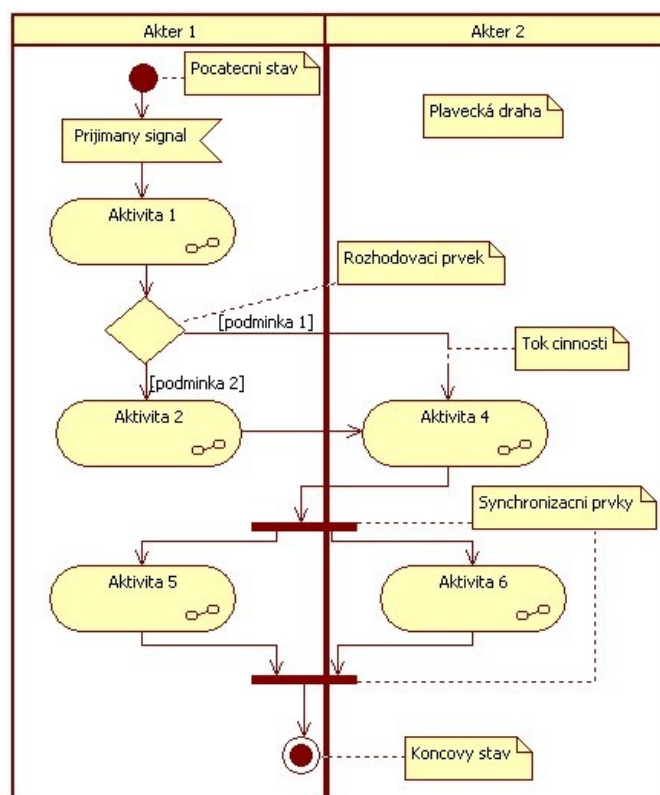
Vztah <<extend>> spojuje případy užití a znázorňuje rozšíření (extend) jednoho případu druhým, tedy že jej může spustit za splnění určitých podmínek.

Ohraničení systému slouží k oddělení funkcí systému od chování prvků vně systému, na které nemá vliv.

Diagram aktivit – aktivity diagram

Slouží k popisu jednotlivých případů užití, kdy znázorňuje všechny provedené činnosti od zahájení aktérem, až po dosažení určitého cíle. Tento diagram zobrazuje, v jakém pořadí budou tyto činnosti plněny, jestli se větví a plní paralelně, nebo je průchod řízen podmínkami. Výhoda diagramu aktivit je, že jej můžeme použít k jakémukoliv modelování procesů a aktivit, proto je ho možno využít

nejen ke specifikaci softwarového systému, ale i například při dokumentování interních procesů společnosti, což splňuje mé potřeby. Na obrázku 2-2 je zobrazena základní specifikace tohoto diagramu.



Obrázek 2-2 Diagram aktivit

Aktér znázorňuje aktivní entitu, může se jednat o člověka, ale i to může být signál jiného systému, který spolupracuje s daným případem užití.

Počáteční stav znázorňuje zahájení činnosti daného případu užití, který může mít i více vstupních bodů, které se zakreslují různými počátečními stavy.

Plavecká dráha (swimlane) přiřazuje vykonané aktivity aktérovi, kterému daná dráha patří. Ten je zodpovědný za vykonávání těchto aktivit.

Přijímaný signál je stav, kdy k pokračování v činnosti je nutný externí příchozí signál, například od aktéra, který se jinak v případě užití nevyskytuje.

Rozhodovací prvek slouží k větvení toku činnosti na základě zadaných podmínek, tedy tok pokračuje tou větví, která splňuje podmínky. Později může dojít ke spojení těchto větví, jak to můžeme vidět na příkladě, není to však předpokladem.

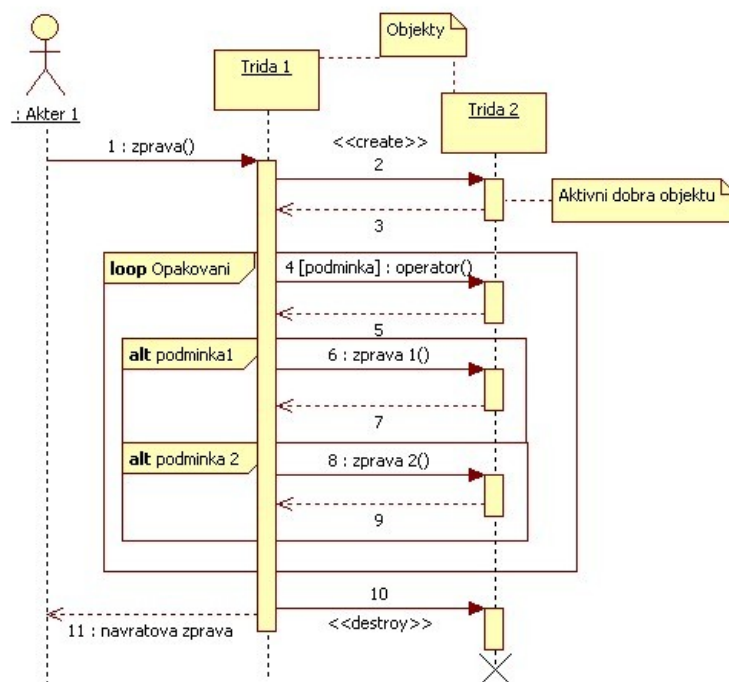
Tok činnosti je časový průchod diagramem, jehož směr určuje posloupnost prováděných aktivit.

Synchronizační prvky také složí k větvení, ale tok činnosti pokračuje do všech částí a dochází tím k synchronizaci následujících aktivit. Tyto prvky musí být vždy v páru a synchronizace není hotová, dokud všechny aktivity v ní nejsou ukončeny.

Koncový stav znázorňuje ukončení činnosti daného případu užití, který může mít i více koncových stavů například v závislosti na úspěšném či neúspěšném dosažení cíle.

Sekvenční diagram – Sequence Diagram

Sekvenční diagram popisuje komunikaci a interakci mezi jednotlivými objekty a to v závislosti na čase. Tok času zde postupuje po svislé ose od shora dolů, kdy dochází ke komunikaci objektů pomocí zpráv, které vedou k naplnění cíle případu užití. Základní prvky použité v tomto dokumentu jsou zobrazeny na obrázku 2-3.



Obrázek 2-3 Sekvenční diagram

Objekty jsou základní prvky, ze kterých je systém sestaven, a reprezentují konkrétní instance tříd systému. Ty mezi sebou komunikují zasláním zpráv a spouštěním metod. Zpráva může obsahovat i podmínku v hranatých závorkách, která povolí odeslání zprávy až po svém splnění. Zprávy mohou být několika typů a to jsou call, return, send, create a destroy.

Zpráva **call** slouží k zavolání jiného objektu, který zpracuje obsah zprávy a navrátí výsledek své činnosti zprávou **return**. Zpráva **send** jen odešle obsah jinému objektu, ale už neočekává návratovou zprávu. Zpráva **create** složí k vytvoření nové instance cílové třídy a naopak zpráva **destroy** ukončí a odstraní cílové objekty.

Aktivní doba objektu je časový úsek, kdy objekt provádí nějakou činnost.

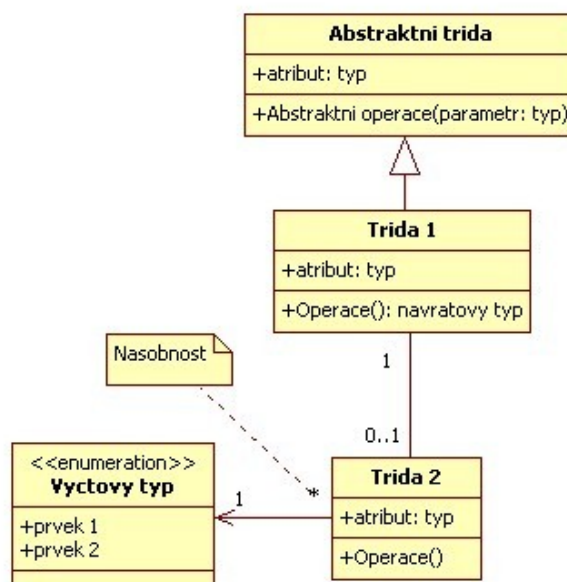
Dalším předmětem sekvenčních diagramů jsou **bloky**, které označují speciální oblasti. Jedná se o bloky **alt** nebo **loop**.

Do bloků **alt** se zobrazují alternativní toky na základě stanovených podmínek, kdy průběh po jejím splnění je zabalen jejím blokem. Těchto podmíněných bloků může být i více a maximálně jeden z nich nemusí znění podmínky obsahovat. Tím jsou v bloku obsaženy všechny jiné situace, než ve vypsáných podmínkách.

Bloky **loop** popisují opakované provádění obsažené části diagramu, pokud je splněna vypsaná podmínka. Jakmile splněná není, tok pokračuje hned za tímto blokem.

Třídní diagram – Class Diagram

Tento diagram je základem z diagramů zobrazující strukturu. Popisuje statický pohled na systém jako soustavu tříd a jejich vztahů, které daný systém tvoří. Ukázkový případ tohoto diagramu je na obrázku 2-4.



Obrázek 2-4 Třídní diagram

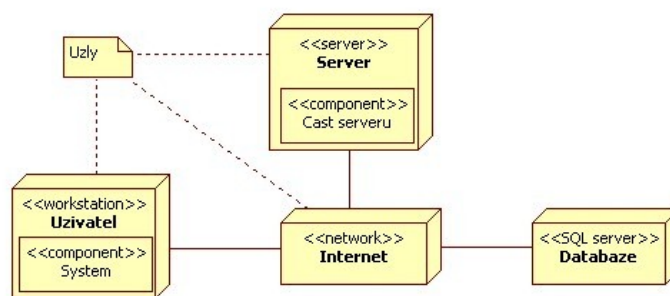
Třída popisuje množinu všech objektů se stejnými charakteristickými vlastnostmi, což jsou atributy metody, relace a chování. Tento prvek je rozdělen na tři menší bloky, kdy do horního se zapisuje název třídy, popřípadě její typ. Do prostředního bloku se zapisuje seznam všech atributů a jejich typů. Nakonec do spodního bloku se vypisuje seznam metod dané třídy, kdy abstraktní metody můžeme označit kurzívou a statické metody podtrženým písmem.

Výčtový typ (enumeration) je speciální typ třídy, který seskupuje množinu logicky příbuzných hodnot.

Asociace zobrazuje množinu spojení mezi objekty, kdy hodnota násobnosti určuje jejich počet. Tato hodnota se udává konstantou (například 1), intervalem (například 1...10) nebo znakem *, který označuje libovolný počet spojení.

Diagram nasazení – Deployment Diagram

Slouží k zobrazení rozložení částí systému na fyzické zařízení a znázorňuje jejich propojení. Je možné jim popsat jak obecný návrh nasazení, tak již zcela konkrétní návrh, který bude obsahovat přímo hardwarové uzly. Obecný náhled na tento diagram je na obrázku 2-5.



Obrázek 2-5 Diagram nasazení

Uzly popisují veškerou fyzickou infrastrukturu jako síť (network), pracovní stanice (workstation), obecné servery (server) či databázové servery (SQL server). Mohou také obsahovat **komponenty** (component), které označují důležitou součást uzlu.

3 Rational Unified Process

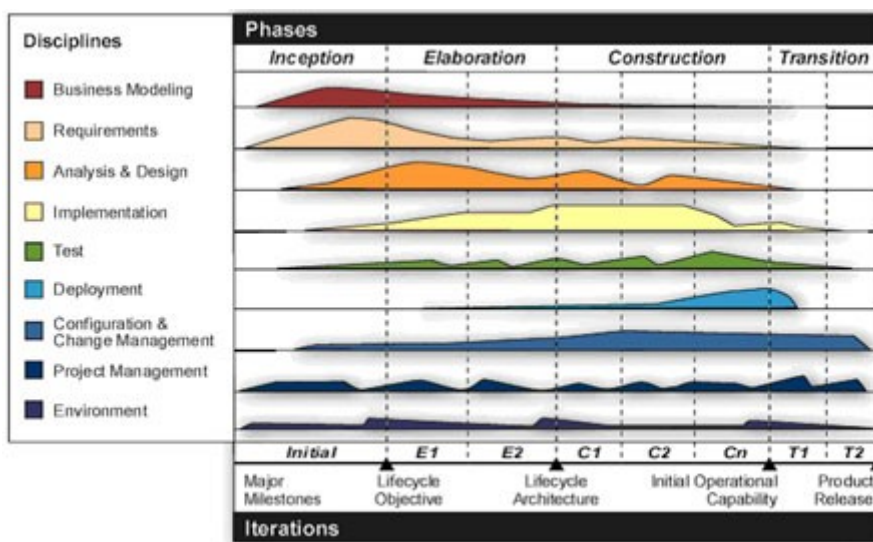
Jazyk UML je kvalitním modelovacím jazykem, ale jak už bylo řečeno, nedokáže definovat metody a postupy pro vytváření systémů, jen je umí pomocí vizuální syntaxe popsat a dokumentovat. Jednou z možností, jak tyto tuto metodiku definovat je použití unifikovaného procesu RUP (Rational Unified Process) [3].

RUP vychází z metodiky Objectory Process [6] a jako první ji začala používat firma Rational Software Corporation [7], která tuto metodiku prezentuje formou softwarového produktu. Jedná se o rámec procesu vývoje softwaru, který popisuje toky činností, které jsou nutné pro vytvoření kompletního a kvalitního softwarového systému. Základem kvalitního systému je samozřejmě splnění požadavků a představ zadavatelů a jeho uživatelů.

Při průchodu jednotlivými toky činností vznikají dokumenty a modely popisující nejen samotný software či systém, ale i znázorňují faktory a vlivy, které působí na jeho vývoj či vyhodnocují jeho uplatnění. V RUP se všechny tyto předměty obecně nazývají **artefakty**. To jaký typ artefaktu zvolit závisí na stavu procesu vývoje a vhodným typem je možno popsat všechny důležité aspekty systému. RUP také zavádí pojem **rolí**, jelikož veškeré činnosti provádí lidské zdroje, je nutné definovat, kdo nese zodpovědnost za správný průběh aktivit procesu a stanovit tak jeho úkoly.

3.1 Fáze vývoje

Každý projekt během svého vývoje prochází určitými fázemi (phases), které mohou obsahovat jednu či více iterací. V různých fázích se klade důraz na různé disciplíny, to vede k tomu, že není nutné například při zahájení projektu sledovat testování nebo naopak v závěru životního cyklu projektu není potřeba se soustředit na byznys modelování. RUP nabízí určitou možnost rozložení objemu prací do čtyř fází, jak můžeme vidět na obrázku 3-1 [3].



Obrázek 3-1 Fáze vývoje

Fáze zahájení (inception) slouží k vymezení požadavků na systém, na němž se shodnou všechny zainteresované strany. Je třeba vymezit rozsah vytvářeného systému a pro klíčové činnosti vytvořit modely případů užití. U projektů, při nichž se pouze upravuje stávající systém, je tato fáze podstatně méně náročná, neboť výše uvedené kroky byly již dříve zčásti provedeny.

Fáze rozpracování (elaboration) má za cíl navrhnout stabilní model architektury systému, který poslouží jako základ pro jeho implementaci. Model architektury vychází ze základních požadavků na systém, vytváří se jeden nebo několik prototypů. Fáze přípravy navazuje na fázi zahájení projektu, na jejím začátku musí být k dispozici úvodní specifikace požadavků a úvodní verze plánu vývoje software zahrnující stručný plán iterací pro tuto fázi.

Fáze konstrukce (construction) je určena k dokončení návrhu, vytvoření a otestování první funkční verze systému. Na rozdíl od předchozích dvou fází je tato zaměřena převážně na implementaci. Základním problémem je efektivní řízení zdrojů - stejně jako v předchozích fázích je vhodné projekt rozdělit na relativně nezávislé části a ty pak přidělit jednotlivým vývojovým týmům. V této fázi projektu by měly být požadavky na systém již stabilní, správa požadavků se už omezuje pouze na zapracování dodatečných požadavků na změny.

Fáze předání (transition) má za cíl předání finální verze systému koncovým uživatelům. V jejím průběhu se provádí beta testování systému a jeho drobné úpravy na základě připomínek uživatelů. V této fázi by již nemělo docházet k žádným zásadním změnám funkcionality software - požadavky zadavatele by se měly týkat pouze instalace, konfigurace a odlaďování drobných chyb zjištěných při testovacím provozu.

3.2 Základní principy

Základní filosofií metodiky Rational Unified Process je šest základních principů používaných při vývoji software: iterativní vývoj, správa požadavků, komponentová architektura, vizuální modelování, průběžné ověřování kvality a řízení změn. Tyto principy byly vyvinuty na základě prakticky ověřených postupů [3].

3.2.1 Iterativní vývoj

Jedná se o rozložení projektu na menší části, které se zpracovávají postupně a pak se postupně na sebe nabalují, dokud není projekt kompletní. Každá iterace musí být uzavřená při přechodu na další, ale je možné provádět změny a úpravy i v částech zpracovávaných v předchozích iteracích, což je oproti vodopádovému modelu velkou výhodou, protože ten je založen na provedení celého systému najednou. V dnešní době rozsáhlých a komplexních systému je téměř nemožné zajistit tak dokonalé provedení toku činností, aby nebylo nutné se zpět vracet. Obecné schéma iterací je znázorněno na obrázku 3-2.



Obrázek 3-2 Iterativní vývoj

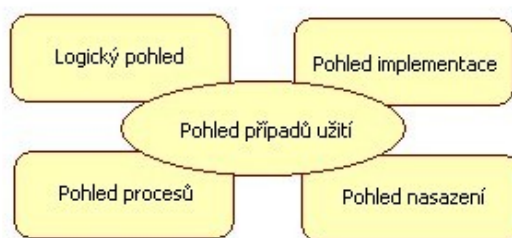
Dalšími výhodami iterativního vývoje je možnost objektivního posouzení stavu projektu, rovnoměrnější pracovní vytížení vývojového týmu, což přímo souvisí s předchozím bodem. Projekt je koncipován tak, aby co nejdříve přinášel konkrétní výsledky. Při rozdělení na iterace je možné snáze sledovat průběh projektu a dodržování stanovených termínů pro jednotlivé iterace. Dále je zde možnost testování mezi verzí, spolupráce s uživateli v průběhu celého projektu, snazší zapracování změn požadavků či včasné rozpoznání nesrovnalostí mezi požadavky, návrhem a implementací. Díky tomu, že uživatelé mají možnost kontrolovat a hodnotit dílčí části systému, značně se omezuje riziko vysokých nákladů způsobených úpravami produktu v pozdní fázi vývoje.

3.2.2 Správa požadavků

Požadavky na systém jsou základními kameny při vytváření softwarového systému, které nemusí jen popisovat požadovanou funkčnost, ale i požadavky na výkon, bezpečnost, rozšiřitelnost apod. U vodopádového přístupu se požadavky sbírají a dokumentují pouze na začátku celého projektu, jejich změna v dalších stádiích vývoje se zpravidla nepřipouští. RUP naproti tomu využívá koncepci “aktivní správy požadavků” spočívající ve stálém kontaktu zadavatele s dodavatelem a možnosti zpřesňování a úprav požadavků v průběhu projektu.

3.2.3 Komponentová architektura

Rozklad složitého systému na jednotlivé komponenty podobné funkcionality není jen aspekt metodiky RUP, ale používá se při vývoji všech robustních architektur. V případě RUP však postupujeme ještě dále a to k dokumentaci nejen samotných komponent, ale i k popisu způsobu, jakým tyto komponenty mezi sebou komunikují, jak se vzájemně ovlivňují a jak jsou zasazeny do příslušného hardwaru. Tomuto se říká pohled „4+1“ („4+1“ view), kde číslovka „4“ znamená čtyři různé pohledy na architekturu a „+ 1“ znamená, že vše zastřešuje a řídí pohled případů užití, jak je znázorněno na obrázku 3-3.



Obrázek 3-3 Pohled 4 + 1

Pohled případů užití zachycuje základní požadavky kladené systém, které určují rozsah a charakter celého systému.

Logický pohled popisuje nejdůležitější třídy a objekty, jejich organizaci v servisových balíčcích a v subsystémech a dále pak organizaci těchto subsystémů. Důležité je zobrazení, jak tyto objekty implementují chování celého systému.

Pohled implementace zachycuje soubory a komponenty tvořící výsledný kód systému a jejich vzájemné závislosti.

Pohled nasazení popisuje fyzické konfigurace, na které pak bude software aplikován. Popisujeme zde fyzické objekty, na kterých bude software spuštěn (CPU, počítač, server) a jejich mezi-komunikační parametry (BUS, internet, LAN).

Pohled procesů je popis jednotlivých procesů a vláken v architektuře. Popisuje organizaci procesů do skupin a také poukazuje na akce mezi jednotlivými procesy.

3.2.4 Vizuální modelování

Primárně je k vizuálnímu modelování navržen jazyk UML, který je využit i v této práci, jak je popsáno v předchozí kapitole. Výhody vizualizace především ulehčují komunikaci, jak mezi analytiky a zadavateli či uživateli, tak mezi členy vývojových týmů.

3.2.5 Průběžné ověřování kvality

Jak už bylo řečeno, základem systému je vysoká míra kvality. RUP umožňuje kontrolu kvality artefaktů během celého životního cyklu vývoje. Každá iterace je podrobena ověřováním kvality, jak v jejím průběhu, tak na jejím ukončení formou prezentací či spustitelných prototypů. To umožní nahlédnout na dopad jednotlivých požadavků a odstranit tak nedostatky, či přidat požadavky nové. Vodopádová metoda toto neumožňuje, protože se u ní provádí testování až na konci životního cyklu projektu.

Hlavní kritéria jsou **funkcionalita, spolehlivost a výkon**. Funkcionalita znamená, zda systém podporuje všechny funkce vymezené v modelu případů užití. Spolehlivost popisuje, zda systém správně řeší určenou skupinu chybových stavů. Výkon je kritérium, zda jsou dostupnost systému a doba odezvy přijatelné.

3.2.6 Řízení změn

Řízení změn je nezbytné pro zachování konzistence systému. Hlavním úkolem je řídit tvorbu nových verzí software, vytvořit systém předávání posledních verzí veškerých výstupů a koordinovat implementaci změn. Všechny požadavky na změny v projektu musí být zaznamenávány a zanalyzovány, aby se předešlo negativním dopadům či rozsahům mimo možnosti realizace. To umožní sledování průběhu těchto úprav a také je zde možnost zpětně vyhledat kdo a kdy změny požadoval, schválil a zavedl.

3.3 Disciplíny RUP

RUP rozlišuje mezi dvěma druhy disciplín: **hlavní** a **podpůrné**. Hlavní disciplíny zajišťují vlastní práci na projektu, tedy přímo generují artefakty, které specifikují vyvíjený systém. Podpůrné disciplíny slouží k jejich řízení a koordinaci, tedy napomáhají ke správnému a efektivnímu průběhu hlavních disciplín.

Mezi hlavní disciplíny řadíme byznys modelování, specifikace požadavků, analýza a návrh, implementace, testování a nasazení. Do skupiny vedlejších disciplín řadíme řízení změn a konfigurací, projektové řízení a správa prostředí [3].

3.3.1 Byznys modelování

Cílem byznys modelování je hledání společného jazyka mezi pracovníky odpovědných za modelování podnikových procesů a softwarovými návrháři, protože softwarový specialista pravděpodobně nemůže znát skutečné fungování obchodních procesů a naopak podnikový specialista zná obchodní procesy z každodenního provádění, ale neví, jaké informace jsou důležité pro vývoj systému. Tato spolupráce může přinést i vylepšení, které se nemusí týkat jen plánovaného systému, ale i nabídnout obchodnímu specialistovi jiný pohled na procesy, což může přinést návrhy na jejich vylepšení.

Byznys modelování se především snaží identifikovat role, procesy a zodpovědnosti. To umožní zasazení dokumentu do skutečného obchodního prostředí, kde jednotlivé role mají určité funkce a zodpovědnosti v procesech.

Sledují se tak tři aspekty nutné k vytvoření byznys modelů a to pohled funkcí, který definuje poskytované služby, pohled specifikace chování, jenž upřesňuje pracovní postupy při poskytování jednotlivých služeb a pohled struktury, který popisuje objekty, které v jednotlivých procesech vystupují. K tomu slouží převážně diagramy případů užití, aktivitní diagramy a třídí diagramy.

3.3.2 Specifikace požadavků

Tato disciplína slouží k nalezení a popisu požadavků uživatelů a zadavatelů projektu. Tyto informace je obvykle velmi obtížné získat, protože sami zadavatelé předem nedokážou podrobně sepsat všechny požadavky. Velikým přínosem je zde iterativní metodiky, kdy uživatelé nalézají chyby až při zhlédnutí prototypu systému, nebo alespoň po prezentaci aktivitních nebo sekvenčních diagramů. Způsobů, jak tyto informace získat je několik, a to buď formou dotazníků, jednání či pohovorem, nebo sledování procesů ve firmě.

Požadavky se dělí na **funkční** a **nefunkční**. Funkční požadavky jsou popisem toho, za jakým účelem se systém zavádí, k čemu bude systém způsobilý. Obvykle se zaznamenávají pomocí diagramů případů užití a dále se mohou specifikovat aktivitními diagramy či sekvenčními diagramy. Nefunkční požadavky obsahují nároky na daný systém a hranice, které musí či nesmí přesáhnout. Ty se zaznamenávají v textové formě.

3.3.3 Analýza a návrh

Analýza a návrh spojují specifikaci požadavků na systém s jeho implementací, tím získáme model chování struktury výsledného systému. Návrh architektury by měl být robustní, ale zároveň flexibilní. Nejdříve je potřeba na základě analýzy případů užití a dalších požadavků nalézt klíčové třídy a vytvořit analytický model systému. Potom je analytický model doplněn o aspekty týkající se implementace a nasazení systému. Je potřeba definovat prvky dle funkcionality, aby je bylo možné implementovat jako komponenty a určit persistentní třídy. Výsledky této disciplíny by měly být tak přesné, aby podle nich bylo možné systém implementovat.

K modelování se využívá hlavně třídí diagramů, sekvenčních diagramů a stavových diagramů.

3.3.4 Implementace

Implementace má za úkol komponenty vymezené v modelu návrhu převést do spustitelné podoby, tedy dochází k psaní zdrojových kódů jednotlivých komponent nezávisle na sobě. V rámci implementace se též provádí integrace komponent a základní testy jejich funkčnosti.

Jelikož základem implementace jsou komponenty, nejdůležitější diagramy v této disciplíně jsou právě komponentní.

3.3.5 Testování

V této disciplíně se testuje jak kvalita jednotlivých implementovaných komponent, tak správná integrace komponent do komplexního systému. Také se ověřuje, zda byly splněny všechny požadavky jak funkční, tak nefunkční.

Jazyk UML nespecifikuje diagramy pro tuto disciplínu, nejčastější je forma textového zápisu zpráv o testování nebo statistickými grafy.

3.3.6 Nasazení

Nasazení systému je hlavní činností prováděnou ve fázi předávání, již před jejím zahájením by však mělo být důkladně naplánováno. Jeho hlavním cílem je poskytnutí software koncovým uživatelům.

Důležitým vizuálním prostředkem je zde diagram nasazení.

3.3.7 Řízení změn a konfigurací

Tato disciplína je nezbytná pro zachování konzistence systému, neboť jejím hlavním úkolem je řídit správu jednotlivých verzí. Důležitý je i sběr požadavků na změny, jejich analyzování, vyhodnocování a schvalování.

3.3.8 Projektové řízení

Řízení projektu je primární podpůrnou disciplínou zajišťující plánování a koordinaci prací na projektu. Významným úkolem této disciplíny je též zajištění potřebných personálních i materiálních zdrojů. RUP přenechává obecné aspekty této disciplíny speciálním metodikám pro řízení projektů.

3.3.9 Správa prostředí

Cílem správy prostředí je zajistit kompletní přípravy, aby mohl úspěšně proběhnout proces vytvoření daného projektu. Patří mezi to jak opatření prostředí pro vývoj, tak v případě potřeby zajištění podpory týmu vývojářů.

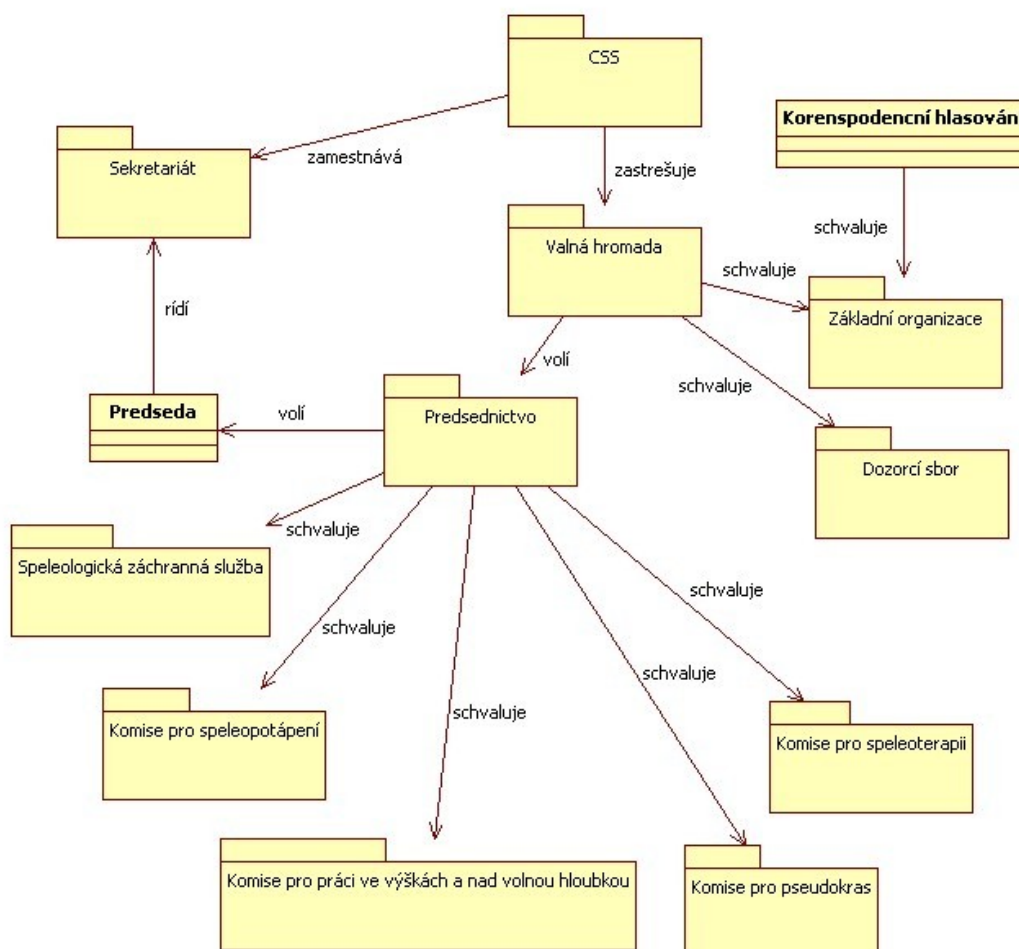
4 Česká speleologická společnost

Česká speleologická společnost [1] je občanským sdružením zájemců o speleologii, krasové i pseudokrasové oblasti a jevy, přirozené i umělé podzemní prostory. Její členové se zabývají objevováním, studiem, dokumentací, průzkumem a výzkumem těchto jevů a prostor, jejich vědeckým výzkumem, ochranou a historickým vyhodnocením. K dalším cílům patří ochrana přírody, shromažďování, uchovávání a využívání získaných poznatků, zveřejňování a popularizace výsledků své práce, rozvoj technického vybavení a prostředků pro speleologii.

ČSS je členem Mezinárodní speleologické unie a spolupracuje s řadou dalších odborných institucí, státními orgány a občanskými sdruženími, jejich činnost souvisí s jejím posláním a činností. Toto sdružení také organizuje různé speciální složky a zastřešuje komise, které řeší specifické otázky z oborů spojenými se speleologií.

4.1 Struktura ČSS

Znázornění základní struktury společnosti můžeme vidět na obrázku 4-1. Je znázorněná jen ta nejnižší úroveň, jednotlivá uskupení se pak dále dělí až na jednotlivé členy s různými funkcemi. Podrobnější znázornění struktury ČSS najdete v příloze č. 1.



Obrázek 4-1 Obecná struktura ČSS

Nyní si jednotlivá uskupení v rámci ČSS můžeme rozdělit do tří kategorií [1]:

- orgány
- organizační jednotky
- pracovní komise

4.1.1 Orgány sdružení

Valná hromada

Valná hromada je nejvyšším orgánem ČSS. Tvoří ji shromáždění delegátů všech jejích organizačních jednotek, členů předsednictva a dozorčího sboru. Valnou hromadu svolává předsednictvo jednou za čtyři roky, nebo mimořádně na žádost dozorčího sboru, nebo na žádost alespoň patnácti procent základních organizací.

Valná hromada schvaluje stanovy a organizační řád ČSS a jejich změny a doplňky, rozhoduje o přijetí a vyloučení ZO jako organizační jednotky ČSS, rozhoduje o rozpuštění ČSS, schvaluje zprávy o činnosti a hospodaření ČSS a dozorčího sboru, stanovuje výši odvodů z členských příspěvků, udílí a ruší čestné členství, volí předsednictvo ČSS, předsedu ČSS a dozorčí sbor na dobu čtyř let.

Předsednictvo ČSS

Předsednictvo ČSS řídí činnost a jedná jménem ČSS v období mezi valnými hromadami. Předsednictvo tvoří předseda, 6 členů a 2 náhradníci.

Předsednictvo ČSS zřizuje a ruší Speleologickou záchrannou službu, jmenuje náčelníka SZS a jeho zástupce, přijímá individuální členy a ruší jejich členství, navrhuje valné hromadě přijetí a vyloučení ZO jako organizačních jednotek ČSS, zřizuje a ruší stanice SZS, svolává a připravuje valnou hromadu, rozhoduje o přidělení dotací a finančních částek z rozpočtu ČSS, řídí a kontroluje činnost sekretariátu ČSS, vydává, upravuje, schvaluje a ruší interní předpisy a směrnice pro činnost jednotlivých složek i celé ČSS, rozhoduje o pozastavení činnosti ZO do rozhodnutí valné hromady.

Předseda ČSS

Předseda ČSS řídí činnost předsednictva podle usnesení valné hromady. Je oprávněn jednat jménem ČSS, případně delegovat tuto pravomoc na jiné členy předsednictva. Předseda ČSS zodpovídá za vypracování výroční zprávy o činnosti předsednictva a činnosti a hospodaření ČSS a její předložení valné hromadě.

Sekretariát ČSS

Sekretariát ČSS je stálý, nevolený administrativní orgán předsednictva ČSS s pracovníky ve stálém pracovním poměru a dobrovolnými pracovníky, schválenými předsednictvem, v počtech stanovených valnou hromadou. Sekretariát za svou činnost odpovídá předsednictvu.

Dozorčí sbor

Dozorčí sbor kontroluje a reviduje činnost složek ČSS dle platných stanov a interních předpisů. Schází se operativně, nebo na žádost organizačních jednotek, nebo jejich orgánů, orgánů a členů ČSS. Provádí výklad stanov a interních předpisů ČSS. Řeší vzájemné spory mezi členy a všemi složkami a orgány ČSS a předkládá předsednictvu a valné hromadě návrhy na řešení. Dozorčí sbor se skládá z předsedy, dvou členů a jednoho náhradníka.

4.1.2 Organizační jednotky sdružení

Základní organizace ČSS

Základní organizace tvoří základní článek ČSS sdružující její členy. Mají právní subjektivitu, mohou svým jménem nabývat práv a zavazovat se. O přijetí a vyloučení ZO jako organizační jednotky ČSS rozhoduje valná hromada na návrh předsednictva.

V současné době má Česká speleologická společnost 63 základních organizací.

Speleologická záchranná služba ČSS

Speleologická záchranná služba je specializovaná složka ČSS pro poskytování pomoci při nehodách v jeskyních, případně v jiných podzemních prostorách. SZS zřizuje a ruší předsednictvo ČSS. Její činnost se řídí vlastním organizačním řádem.

Činnost SZS řídí náčelník SZS, který je oprávněn jednat jménem ČSS ve věcech SZS a zodpovídá za vypracování výroční zprávy o činnosti SZS a její předložení předsednictvu ČSS ke schválení. Náčelníka SZS jmenuje a odvolává předsednictvo. Akčním článkem SZS jsou stanice SZS, které samostatně provádějí zásahy při nehodách v jeskyních, případně v jiných podzemních prostorách. O vzniku a zrušení stanic SZS rozhoduje předsednictvo na návrh náčelníka SZS.

4.1.3 Pracovní komise

Komise pro speleopotápění

Komise pro speleopotápění byla zřízena předsednictvem ČSS pro řešení specifických otázek v oblasti speleopotápění. Komise odpovídá za organizaci výcviku, zkoušení a udělování kvalifikací pro speleopotápění v rámci ČSS, vede evidenci udělených kvalifikací a jmenuje instruktory výcvikového systému. Dále shromažďuje informace o speleopotápěčských lokalitách a koordinuje spolupráci speleopotápěčů se Speleologickou záchrannou službou ČSS.

Komise pro speleoterapii

Komise pro speleoterapii byla zřízena předsednictvem ČSS pro řešení speciální problematiky speleoterapie a spolupráce se zahraničními partnery a s Mezinárodní speleologickou unií.

Komise pro pseudokras

Komise pro pseudokras byla zřízena předsednictvem ČSS pro řešení speciální problematiky pseudokrasu a spolupráce se zahraničními partnery a s Mezinárodní speleologickou unií. Komise koordinuje organizování akcí s pseudokrasovou problematikou a centralizuje výsledky činnosti jednotlivých ZO ČSS pracujících v pseudokrasových oblastech.

Komise pro práci ve výškách a nad volnou hloubkou

Komise pro práci ve výškách a nad volnou hloubkou byla zřízena předsednictvem ČSS a její náplní je sledování vývoje a změn v legislativě týkající se speleoalpinismu a prací ve výškách a nad volnou hloubkou, soustřeďování informací o současných i nových pomůckách a materiálu pro tuto činnost a zkušeností s jejich používáním a poradenství v tomto oboru pro členy ČSS formou kvalifikovaných odpovědí na konkrétní dotazy.

4.2 Základní názvosloví

Pro pochopení některých struktur a postupů je nutné znát význam všech označení a chápat, jaké aktivity výraz zastupuje, proto jsem popsal méně známé výrazy použité ve specifikaci ČSS.

Pseudokrasem [8] označujeme soubor podzemních i povrchových tvarů a jevů, které jsou morfologicky obdobné tvarům a jevům krasovým, ale vznikly převážně jinými procesy než rozpouštěním v různých typech sedimentárních, vyvřelých i metamorfovaných hornin. Hlavními modelujícími pochody jsou řícení a rozpad masivu, mechanická činnost proudící vody nebo větrů, vyplavování částic z pozemí apod.

Speleoterapie [9] má dva druhy významu. V prvním případě se jedná o označení interdisciplinárního vědního oboru, který se vyvinul na hranici mezi přírodními a lékařskými vědami. Zabývá se strukturou a dynamikou jeskynního prostředí, režimem jeho složek a zároveň studuje míru a způsob jejich vlivu i důsledky jejich působení na lidský organismus. V druhém případě se rozumí soubor léčebných a doplňkových metod, stanovených na základě poznatků teoretické speleoterapie a využívajících specifických vlastností přírodního prostředí krasových jeskyní i některých umělých podzemních prostor (důlních děl) k léčbě určitých onemocnění nebo k omezení či dokonce odstranění poruch funkcí některých systémů lidského organismu.

Prolongace [10] v obecném pojetí znamená prodlužování. Ve speleologii je význam podobný, jedná se o manuální prodlužování, zvětšování prostor nebo zpřístupnění prostor nových. Nejčastěji se jedná o práce výkopové, tedy odstraňování volných sutí, naplavenin či úlomků masivu.

Explore [10] je postup vědeckého bádání, kdy se na základě podrobného studia jevu v konkrétních a přirozených podmínkách dochází k pojmům a k hypotézám. Ve speleologii se směr bádání zaměřuje na podzemí, ať se jedná o jeskyně přírodního charakteru, nebo díla vzniklá lidským přičiněním. Základním výstupem explorační speleologie jsou nabyté vědomosti, které se v další fázi (mapování) převádějí do digitální (záznam v osobním počítači) nebo analogové (záznam na papír) podoby.

Chiropterologická fauna [11] taky jinak nazývaní letouni, jsou jedinými savci schopnými aktivního letu a to za pomoci předních končetin přeměněných v křídla. Známa je i schopnost letounů upadat v nepříznivých zimních podmínkách do stavu strnulosti (hibernace) nebo jejich orientace pomocí ultrazvukových signálů (echolokace).

5 Systém pro ČSS

Základy teorie a struktury firmy jsme si stanovili v minulých kapitolách, ale primárním cílem tohoto dokumentu je popsat procesy vykonávané Českou speleologickou společností za účelem jednoznačného popisu základních postupů a stanovení možností na jejich vylepšení. Potom na těchto poznatcích postavit analýzu a návrh systému.

Předem nutno říci, že ČSS prozatím nemá žádný komplexní systém, který by sloužil k evidenci a správě všech jejích prvků. Taky je potřeba dodat, že se jedná o neziskovou organizaci, která nemá své cíle v uspokojování potřeb zákazníka, čili veškeré postupy by měly vést převážně k zefektivnění jejich dosavadních procesů.

Protože se jedná o velice rozsáhlou společnost, se složitou strukturou a spoustou procesů, rozhodl jsem se po dohodě se zadavateli, nastavit priority zpracování jednotlivých procesů na základě jejich umístění v organizační struktuře.

5.1 Nastavení priorit

Základem struktury jsou orgány, organizační jednotky a pracovní komise, jak můžeme vidět v předchozí kapitole, které pak obsahují jednotlivá uskupení. Nastavení priorit procesů jednotlivých částí je vyznačeno v následující tabulce. Čím vyšší hodnota, tím vyšší priorita.

Typ uskupení	Uskupení	Priorita jejich procesů
Organizační jednotka	Základní organizace	12
Komise	Pro práci ve výškách a nad volnou hloubkou	11
Speciální typ	Korespondenční hlasování	10
Komise	Pro pseudokras	9
Orgán	Valná hromada	8
Orgán	Předsednictvo	7
Orgán	Předseda	6
Orgán	Dozorčí sbor	5
Orgán	Sekretariát	4
Organizační jednotka	Speleologická záchranná služba	3
Komise	Pro speleoterapii	2
Komise	Pro speleopotápění	1

Přechod na nižší prioritu se bude provádět nejdříve až v následující iteraci. Jednotlivá uskupení sice navzájem spolupracují, ale jinak jsou od sebe odděleny a mohou existovat samostatně.

5.2 Nový systém – 1. iterace

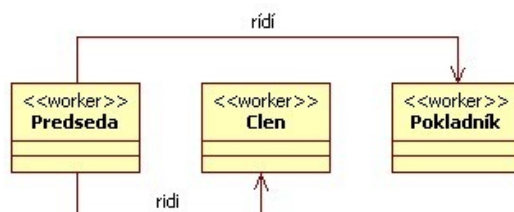
Na základě tabulky priorit je patrné, že nejvyšší prioritu procesů má organizační jednotka sdružení Základní organizace, kterou v následujících odstavcích podrobně prozkoumám a zdokumentuju jednotlivé požadavky a aktivity tohoto uskupení.

5.2.1 Byznys modelování – 1. iterace

Obecná definice Základní organizace je již v kapitole 3. Česká speleologická společnost, ale pro další průzkum problematiky jejich procesů je nutné popsat její strukturu až na základní prvky. Informace byly získány především z dotazníků a briefingu. Tyto dokumenty je možno najít v příloze č. 2.

Organizační struktura ZO

Základní organizace jsou uskupení lidí – členů, která vznikají dle lokální orientace uskupení, jsou tedy spojeny s určitým místem. Jejich struktura je obecně stejná, skládá se z předsedy, pokladníka a řadových členů jak je znázorněno na obrázku 5-1. Ještě zde existují speciální případy jako místopředseda, ale ten se navenek projevuje jako člen a v potřebných situacích jako předseda. Další speciální případ je čestný člen, ale tady nejde o jinou roli, jen o titul vydávaný za záslužnou činnost.



Obrázek 5-1 Organizační struktura ZO

Předseda vede ZO a je obvykle i jejím zakladatelem. Jeho úkolem je řídit úsilí a práci jednotlivých členů, je hlavním organizátorem veškerých činností v rámci ZO, zastupuje ZO ve sporech a spravuje administrativní záležitosti v ZO. Jeho základní procesy jsou znázorněny na následujícím obrázku 5-2.



Obrázek 5-2 Základní procesy předsedy ZO

Pokladník je osoba, která obstarává finanční plánování ZO a zodpovídá za vyhotovení rozpočtů na akce a expedice. Jeho základní procesy jsou znázorněny na následujícím obrázku 5-3.



Obrázek 5-3 Základní procesy pokladníka ZO

Člen je základní prvkem, který vykonává primární procesy, ke kterým byly ZO založeny. Tyto procesy jsou znázorněny na následujícím obrázku 5-2.



Obrázek 5-4 Základní procesy člena ZO

Specifikace hlavních procesů ZO

V tomto odstavci se zaměříme na modelování nejdůležitějších procesů ZO, které musí být provedeny pro úspěšné splnění požadovaného úkolu. Popis procesů tak umožňuje řídit a monitorovat jednotlivé prováděné činnosti. V případě potřeby můžeme procesy optimalizovat či navrhnout úpravy pro efektivnější využití zdrojů. Ke každé specifikaci je přiložen přehledový diagram toku činností daného procesu.

Základní procesy jsou ty, které se provádějí často v krátkém časovém úseku a jsou základními náplní členů ČSS. Patří mezi ně evidence členů ZO, explorace a prolongace jeskyní, mapování a sledování chiropterologické fauny.

Evidence členů ZO je nutná nejen kvůli kontaktním údajům a kvůli snadnější orientaci ve struktuře ZO, ale taky kvůli odvádění příspěvků ČSS za člena. Evidence obsahuje informace o jménu, příjmení, adrese, data narození, e-mail adresa, popř. jiný kontakt a taky příslušnost k ZO, protože se pak údaje na sekretariátě centralizují. Nový člen se zavádí do evidence dle vyplněné přihlášky, stávající jen nahlásí změny předsedovi ZO. Pokud byl člen vyloučen nebo sám vystoupil z členství k ZO, je smazán z evidence. Informace se ukládají do Excel dokumentu.

Primární aktér: Předseda ZO

Rozsah: Základní organizace

Interní aktéři: Člen ZO

Vstupní podmínky: Předseda ZO má vytvořenou tabulku v programu Excel, do které vkládá údaje o členech. Člen musí souhlasit s přístupem sekretariátu a předsedy ZO k jeho osobním informacím. Předpokladem zápisu člena do evidence je čekací zkušební lhůta.

Záruky úspěchu: Informace o členovi byly úspěšně předány na sekretariát.

Hlavní scénář:

1. Člen ZO změnil své kontaktní údaje.
2. Člen ZO nahlásí změny Předsedovi ZO.
3. Předseda ZO vyhledá člena ZO v evidenci.
4. Předseda ZO provede zápis údajů o členovi ZO na základě získaných informací.
5. Předseda ZO odešle evidenci na sekretariát.

Vedlejší scénář:

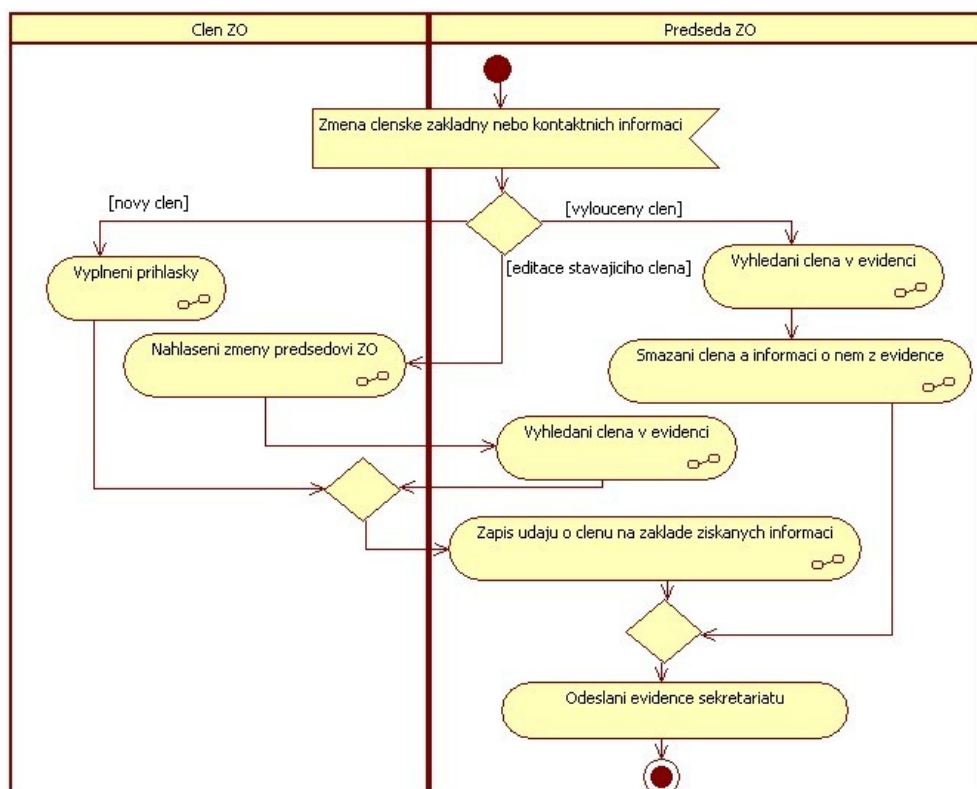
- 1a. Nový příchozí člen ZO.

Nový člen vyplní přihlášku a tu předá předsedovi ZO. Scénář pak pokračuje standardně krokem 4.

- 1b. Vyloučený či odstupující člen ZO.

Předseda ZO vyhledá člena v evidenci a pak smaže veškeré informace o něm. Scénář pak pokračuje standardně krokem 5.

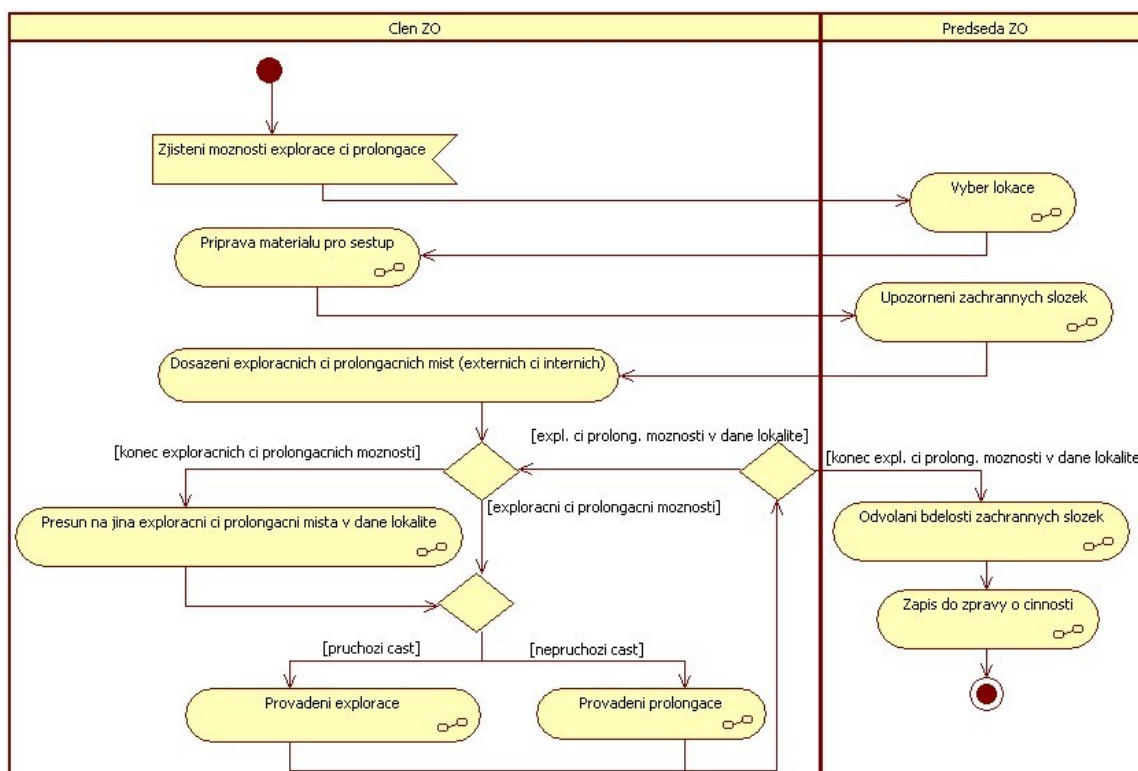
Diagram toku činností procesu (obrázek 5-5):



Obrázek 5-5 Diagram aktivit: Evidence členů ZO

Explorace a prolongace jeskyní popisuje hledání nových podzemních prostor, popřípadě ruční zvětšování těch nalezených. Zvláštností u těchto prací je nutnost upozornění záchranných složek, protože je tady veliké riziko zranění či ohrožení života. Explorace se provádí průchodem daného objektu, pokud už není dále průchodný, začíná prolongace, čili prodlužování objektu. Ta se provádí povětšinou manuálně popřípadě pomocí elektrického nářadí. Po ukončení prací je nutné odvolat obezřetnost záchranných složek a postupy slovně popsat do zprávy o činnosti.

Diagram toku činností procesu (obrázek 5-6):



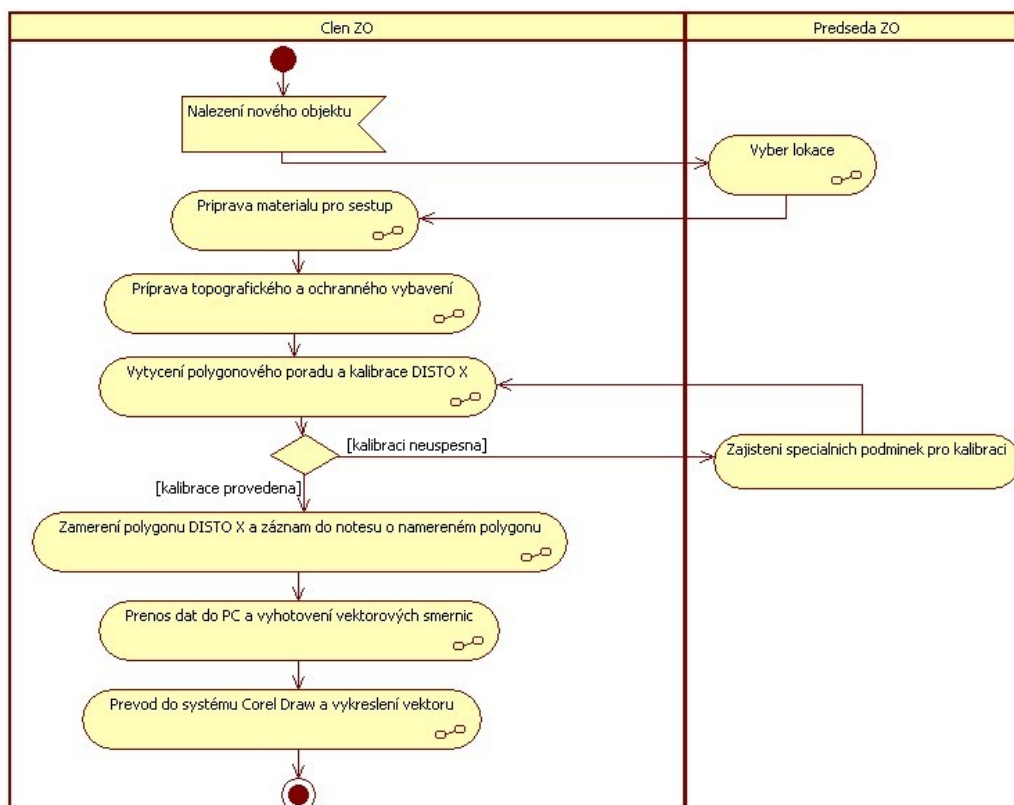
Obrázek 5-6 Diagram aktivit: Explorace a prolongace jeskyní

Mapování slouží k zakreslení objektů do 2D obrazu. Používá se k němu polygonových terčíků, upravený vzdálenostní měřič Disto X [12], zápisník k záznamu informací o jednotlivých měřeních a laserové ukazovátko. Disto X je nutno před každým mapováním kalibrovat, aby hodnoty byly přesné. Jednotlivá měření se zapisují, protože přístroj neumožňuje ukládat další informace o měření, jen identifikační číslo, vzdálenost, sklon a zeměpisnou orientaci. Po zaměření se tyto hodnoty převedou do osobního počítače a tam se pomocí programu ToPo převedou náměry na vektory. Pomocí zápisníku se jednotlivé vektory posunou z nulového bodu do sítě polygonů. Tato síť se převede do programu Corel Draw, kde se dokreslí tvar objektu (odhadovaný dle měření řezů).

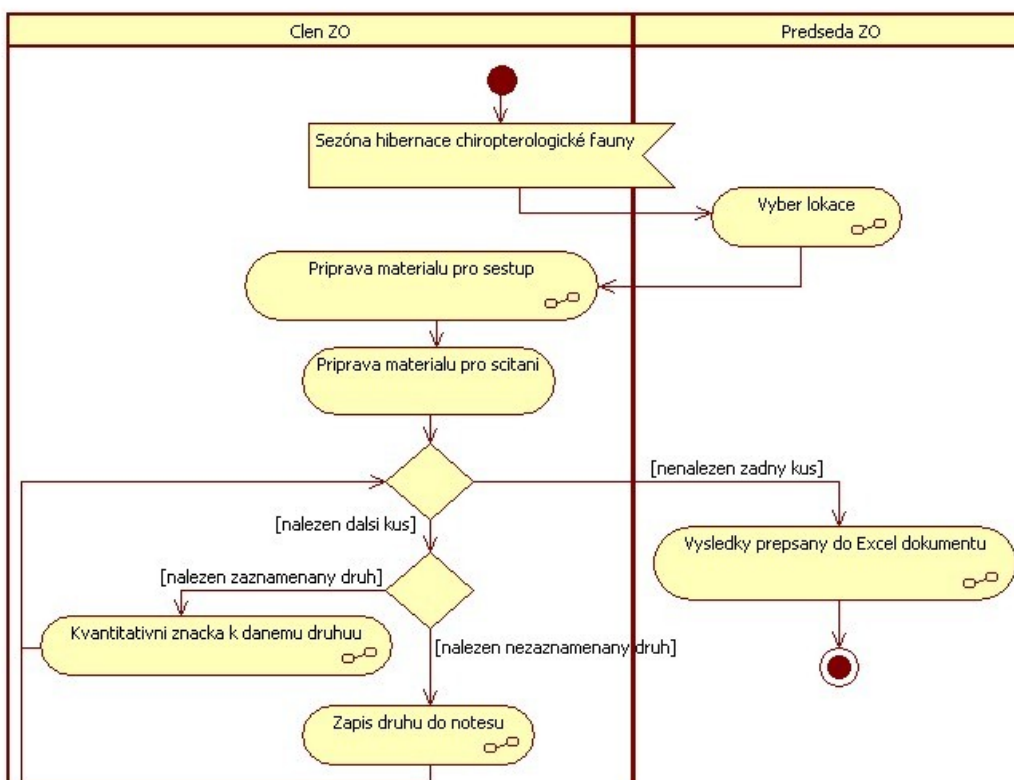
Diagram toku činností procesu (viz obrázek 5-7).

Sledování chiropterologické fauny se provádí pohledovým zkoumáním při jejich hibernaci v podzemí v zimních měsících. Hledají se jednotlivé druhy, každý druh je zapsán do notesu a k němu pak člen ZO zaznamenává počet nalezených kusů v průběhu sledování, tím se zajistí, že dokument obsahuje jen nalezené druhy. Výsledky sledování jsou pak převedeny do tabulky programu Excel.

Diagram toku činností procesu (viz obrázek 5-8).



Obrázek 5-7 Diagram aktivit: Mapování



Obrázek 5-8 Diagram aktivit: Sledování chiropterologické fauny

Podrobnější specifikaci těchto procesů a obecnou specifikaci vedlejších procesů, jako výběr lokality, příprava materiálu pro sestup, příprava a plánování akcí v terénu, příprava mezinárodní expedice, příprava výroční členské schůze, výběr členských příspěvků a finanční plánování expedice je možno najít v příloze č. 3.

Návrhy na vylepšení těchto procesů

Při studování a modelování jednotlivých procesů jsem narazil na některé nesrovnalosti nebo nevhodně volené postupy. Většina z nich je způsobená neexistujícím systémem této společnosti nebo tím, že praktiky byly zavedeny v šedesátých letech minulého století a až na pár maličkostí se od té doby neměnily.

Složitá komunikace mezi jednotlivými uskupeními ČSS

Veškeré předávání informací v rámci ČSS probíhá formou elektronické pošty nebo telefonního spojení a tady hned narážíme na několik problému. Hlavním problémem je šíření takto podané informace nebo dat, která se prvotně nedostanou k všem zájemcům, ale k těm účastníkům, které odesílatel považuje za vhodné. Potencionální zájemci, které odesílatel nezvolil, se musí sami na odesílatele kontaktovat a zažádat jej o dané informace. Zde se ovšem objevuje druhá část tohoto problému, protože zájemce o jistou tematiku nemusí vědět, že byly vydány nové informace, popřípadě neví u koho tyto informace hledat. Sekretariát ČSS sice seznamy nových poznatků a publikací vlastní, ale tímto se jednoduchý úkon protahuje na mnoho kroků, které trvají minimálně dny.

Tento problém by měl řešit právě nový systém, který bude portálem pro sdílení dat, informací a komunikace mezi jednotlivými uskupeními. Nové informace se zde budou hned ukládat do seznamů a bude zde také přístup k jednotlivým dokumentům. Podmínkou přístupu k těmto informacím bude přihlašovací účet v tomto systému.

Omezení datové kvantity při předávání informací

Jak už bylo řečeno, k veškeré komunikaci se používá buď telefonů nebo elektronické pošty. Klasické analogové telefony neposkytují žádné jiné přenosové možnosti mimo ty verbální, mobilní telefony jsou omezeny na stovky kilobajtů a elektronická pošta maximálně převede dvě desítky megabajtů, ale složité mapy několikakilometrových systému mohou být znatelně větší.

Přenos dat bude omezen jen fyzickou velikostí disku na databázovém serveru, na který se bude systém odkazovat pomocí seznamu a protokolem FTP (File Transfer Protocol) bude k jednotlivým dokumentům přistupovat.

Složitě přemísťování seznamu členů

V procesu evidence členů jsem narazil na zvláštní systém zavádění změn do seznamu členů. Každé ZO má svůj vlastní seznam členů (obvykle uchovávaný v elektronické podobě dokumentu Excel) a pak existuje kompletní seznam členů, který vlastní sekretariát. Kdykoliv dojde k nějaké změně, předseda ZO ji zaneše do svého interního seznamu a ten pak celý přeposílá na sekretariát, kde dojde buď k přepsání celé oblasti daného ZO, nebo ke složitému vyhledávání změn.

Hlavní výhodou systému bude ucelená evidence všech členů, kdy na základě rolí budou rozdány oprávnění a přístupy jednotlivým uskupením a jejich členům. Tato evidence bude uložena v databázi, takže šíření změn bude téměř okamžité.

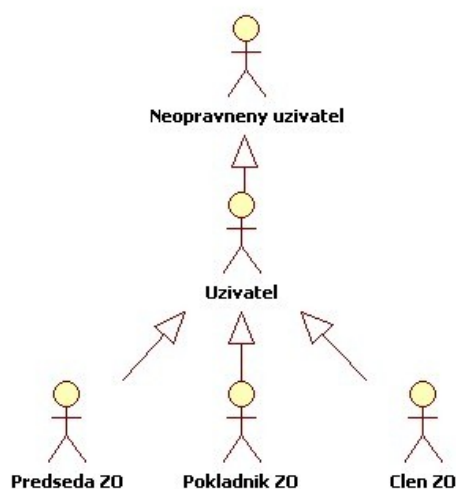
Propagace událostí

Kvalitní propagace je základem úspěšné akce nebo expedice. Z důvodu nedostatku členů může být akce úplně zrušena, nebo naopak ve většině aktivitách členů ČSS neexistují početní limity. Využívání mobilní sítě k propagaci je sice účinné, protože dotyční se k informacím dostanou velmi rychle, ale taky to obsahuje jisté finanční náklady. Jelikož je ČSS nezisková organizace i menší výdaje si hradí členové sami. Další forma je elektronická pošta, tady je ale nevýhoda, že nelze jednotlivé nabídky nějak systematicky třídit.

Systém může sloužit jako databáze těchto aktivit, ale také může mít zpětnou vazbu, kdy na nabídky mohou členové reagovat a k jednotlivým událostem se hlásit, což urychlí i práci předsedovi ZO, který rychlým zhlédnutím zjistí, zda je akce realizovatelná, nebo je lepší ji zrušit.

5.2.2 Specifikace požadavků – 1. iterace

Na základě rozboru procesů a návrhů na jejich vylepšení můžeme stanovit specifikaci požadavků na nový systém. V první iteraci se zaměříme na základní kostru systému se zabudovanou evidencí členů a budou zde vystupovat jen role oprávněného uživatele (obecně ho nadále budeme nazývat uživatelem) a neoprávněného uživatele. Strukturu rolí zobrazuje obrázek 5-9.



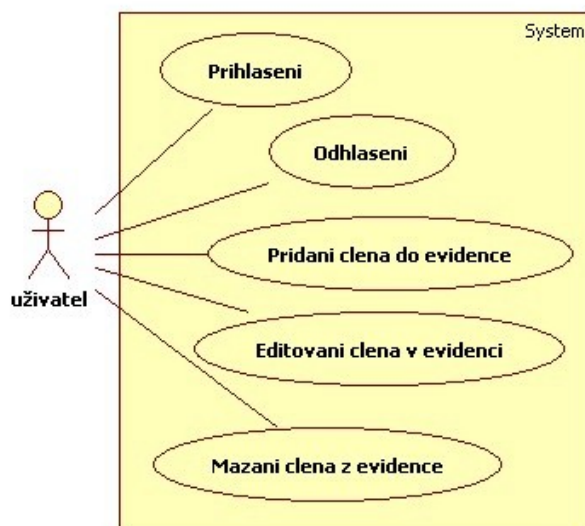
Obrázek 5-9 Struktura rolí

Neoprávněný uživatel je kdokoliv z široké veřejnosti, kdo má přístup k systému, ale nemá práva pro vstup do něj. Tento uživatel nemá žádnou možnost se dostat k informacím a umožnění práce se systémem musí projít registrací jako řádný člen ČSS. Po přijetí jej do systému zavede odpovědný uživatel.

Uživatel má přístup do systému a to znamená, že jeho registrační údaje jsou již v systému zavedeny. Tato role zapouzdřuje jednotlivé typy účastníků ZO: předseda, pokladník a člen.

Funkční požadavky

Nyní můžeme na základě rolí definovat požadavky kladené na systém. Bude zde vystupovat jediný aktér uživatel, který bude zastupovat všechny účastníky ZO. Jednotlivé funkce budou definovány jako případy užití, jak je zobrazeno na obrázku 5-10. Podrobnější modelování jednotlivých požadavků je k nalezení v příloze č. 4.

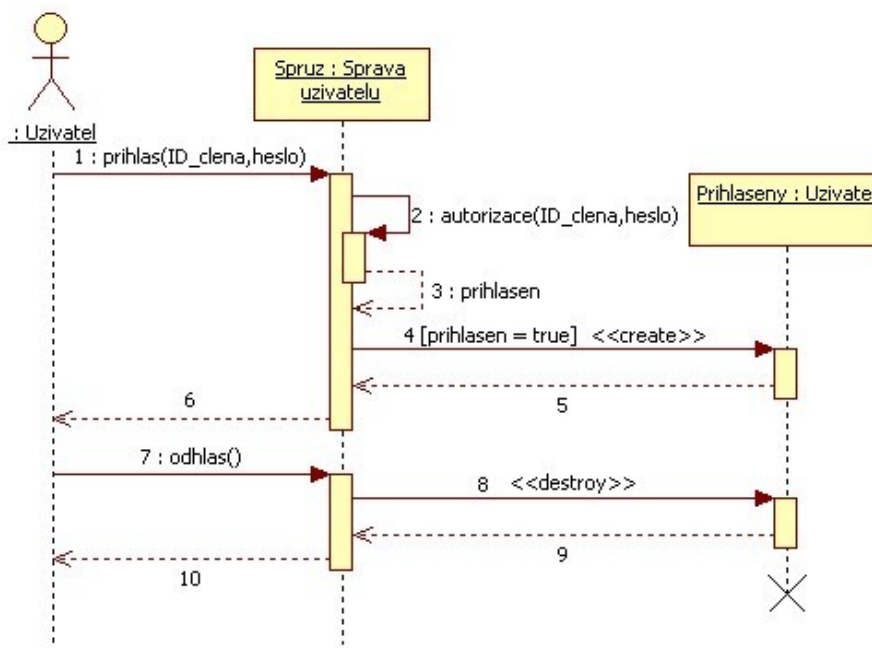


Obrázek 5-10 Diagram případů užití - Funkční specifikace požadavků

Přihlášení je nutnou podmínkou k práci se systémem. Každý uživatel bude mít přiřazené unikátní číslo a zvolí si heslo pro přístup do systému. Unikátní číslo se skládá z prvních třech písmen příjmení uživatele a trojčíferného čísla, které zajistí unikátnost v případě podobných nebo stejných příjmení. Heslo bude možno změnit daným uživatelem.

Odhlášení slouží uživateli k opuštění systému, aby nikdo další nemohl zneužít jeho přístupových práv, ale zároveň není nutné systém znova inicializovat a další uživatel se může do systému přihlásit.

Sekvenční diagram popisující přihlášení a odhlášení uživatele je znázorněn na obrázku 5-11.

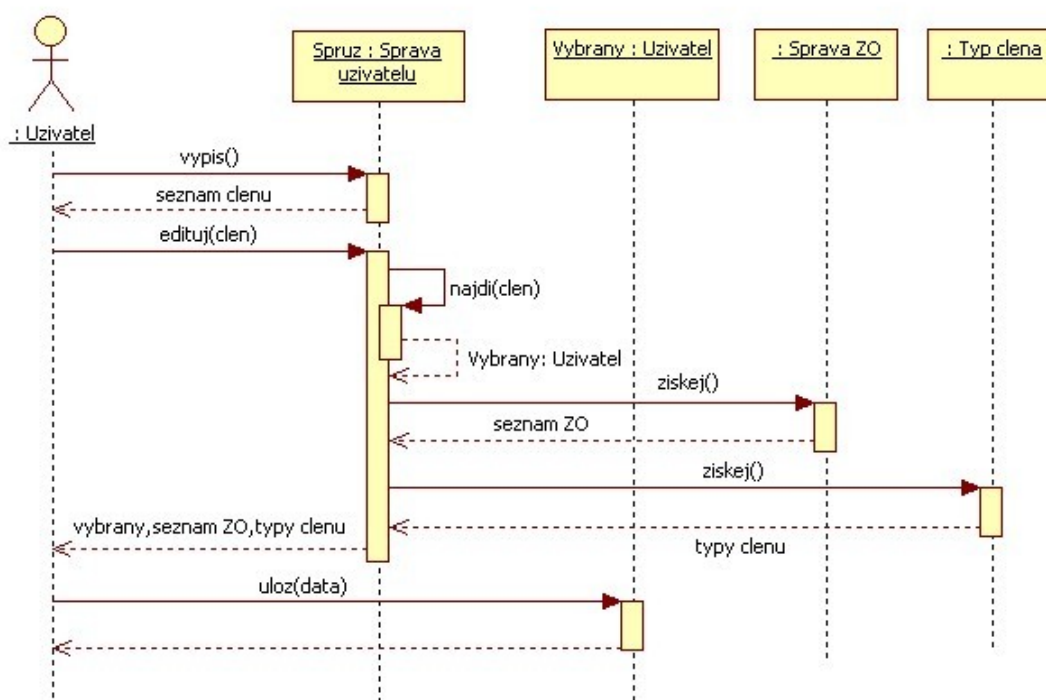


Obrázek 5-11 Sekvenční diagram: Přihlášení a odhlášení

Přidání člena do evidence je operace, kterou může provést jen přihlášený uživatel. Cílem je zavedení nového člena do systému, který se tak dostane do role uživatele. Důležité je přiložení všech informací o novém členovi, aby mohly být v případě potřeby vyhledány a použity k potřebným úkonům, proto není možné vytvořit uživatele bez kompletního vyplnění formuláře.

Editování člena v evidenci zajišťuje možnost úpravy údajů o uživateli. Je dnes běžné, že lidé mění své bydliště a jelikož mezi členy ČSS patří i svobodné ženy, dá se očekávat, že dojde ke změně jejich příjmení. Nejdůležitější částí jsou však telefonické nebo emailové kontakty, které v systému musí být aktuální a přesné, protože jsou základem komunikace jednotlivých uživatelů.

Jelikož editování člena v evidenci obsahuje prvky jak přidávání, tak mazání členů, je zobrazen právě tento sekvenční diagram jako zástupce, pro představu komunikace objektů na obrázku 5-12.



Obrázek 5-12 Sekvenční diagram: Editování člena v evidenci

Mazání člena z evidence má za úkol nejen udržovat aktuální informace o počtech členů, ale také splňuje právní podmínky podobných evidencí. Obsahují totiž citlivé informace, které není možno zveřejňovat a každý člen, který vstupuje do ČSS podepisuje dokument, který povoluje uchovávání jeho osobních údajů, ale po vystoupení z organizace ztrácí tento dokument platnost a proto je nutné informace nenávratně odstranit.

Nefunkční požadavky

Tato část je neméně důležitá jako požadavky funkční, protože poukazuje na základ obecných podmínek, kladených na výsledný systém. Nefunkční požadavky se zabývají použitelností, spolehlivostí, výkonem, rozšiřitelností a udržitelností systému. Také je nutné dodržet určité standardy, které sice technologie vývoje systému nevyžaduje, ale vyžadují je nařízení a normy společnosti, která bude systém využívat. Tyto požadavky byly sepsány na základě domluvy se zadavateli.

Základní požadavky:

- Systém bude přístupný 24 hodin denně 7 dní v týdnu. Případné pády systému nebudou převyšovat 4% veškerého času.
- Produkt bude přístupný pomocí aplikace spustitelné na osobním počítači s internetovým připojením.
- Uživatelské rozhraní bude vytvořeno pro lehké použití pro uživatele se základními počítačovými znalostmi a nebude vyžadovat další učební procesy.

Aplikační standardy:

- Uživatelský interface bude kompatibilní s Windows95/98/NT/XP/Vista.

Požadavky na systém:

- Systém bude operovat na osobním počítači P333 nebo lepším.
- Systém nebude požadovat více jak 128 MB RAM a 1 GB Diskový prostor.
- Komponenty systému budou spustitelné na operačních Windows95/98/NT/XP/Vista.

Požadavky na výkon:

- Systém musí být schopen obstarávat až 100 uživatelů ve stejný čas.
- Systém zaručuje, že prodlení přístupu k jakékoliv databázi nebude větší než 30 sekund.
- Systém musí obstarat 80% transakcí do 2 minut.

5.2.3 Analýza – 1. iterace

V této části je třeba analyzovat základní doménu a nalézt tak nejdůležitější třídy objektů, jejich atributy, zodpovědnosti a vazby, které musí být voleny tak, aby splnily funkční požadavky definované v minulé kapitole.

Třídy objektů

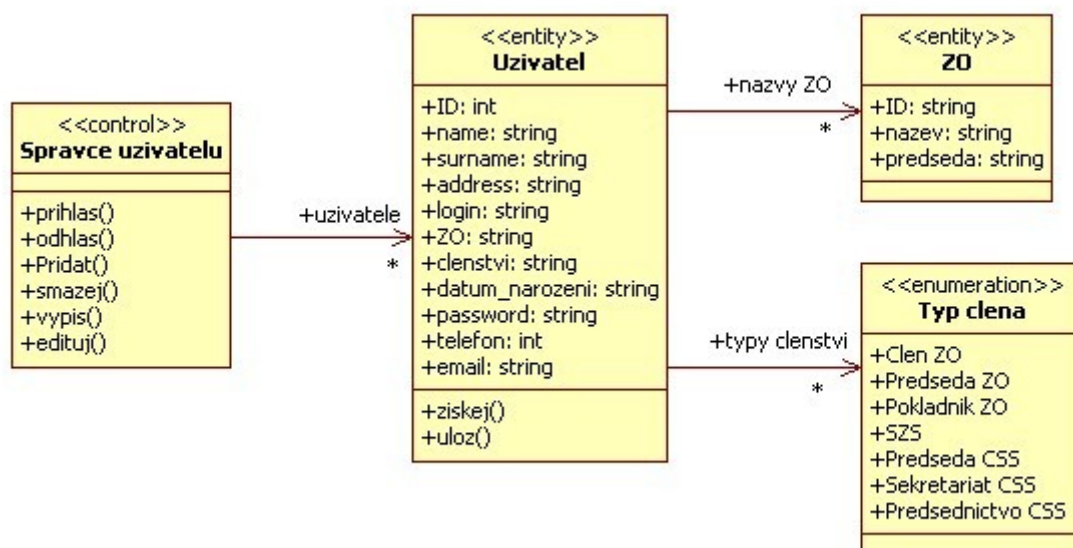
Třídy objektů popisují skutečné objekty problémové domény a byly nalezeny pomocí modelování byznys procesů a specifikace požadavků. Nyní provedeme jejich detailnější popis a rozbor a tím zjistíme jejich vnitřní strukturu. Základní diagram třídních objektů je zobrazen na obrázku 5-13.

Nejobjemnějším balíčkem v hlavní části je objekt **Uživatel**, který obsahuje všechny možné informace a funkce pro vedení evidence uživatelů. Je potřeba evidovat nejen jejich kontaktní údaje (jak už bylo několikrát zdůrazněno, tyhle údaje jsou základem veškeré komunikace v rámci ČSS), ale také příslušnost uživatelů a jejich postavení ve struktuře společnosti (toto bude užitečné v příštích iteracích, kdy bude nutné stanovit více rolí pro systém).

Obecný balíček **Správce uživatelů** slouží k popisu uživatelů systému, zajišťuje jejich autentizaci při přihlašování a také možnost odhlášení. Také obsahuje hlavní aktivační funkce pro vedení evidence.

Balíček **ZO** je v této iteraci jen zástupce seznamu základních organizací. Jeho struktura je už ale rozšířená a později bude na této struktuře zavedený balíček pro správcování ZO.

Část **Typ člena** je pouhým seznamem (enumeration) jednotlivých postavení členů v ČSS, který slouží k používání přesného a společného názvosloví při vkládání nového uživatele.



Obrázek 5-13 Třídní diagram: Základní struktura objektů

5.2.4 Návrh – 1. iterace

Nejdříve bude nutné zabývat se tvorbou architektury systému a potom provedeme popis základních schopností systému, na které je nutno brát při vývoji zřetel. Tyto schopnosti vycházejí jak z požadavků kladených na systém, tak z obecných pravidel. Na základě analýzy systému vytvoříme už konkrétní objekty a popíšeme je v základní struktuře.

Architektura systému

Architektura systému je zároveň jeho základní kostrou, která na sebe postupně nabaluje všechny funkčnosti aplikace. Její definice by měla být tak podrobná, aby popsala jak funkční, tak nefunkční požadavky, ale také by si měla zachovat flexibilitu, rozšiřitelnost a hlavně možnost spravování.

Jak napovídá probraná teorie, navrhl jsem používání kompozitní architektury 4+1. Pohled případů užití je již rozebrán v předchozí kapitole 5.2.2, proto není nutné se k němu vracet, ale je potřeba rozebrat zbývající pohledy.

Logický pohled

Popisuje nejdůležitější třídy a objekty, jejich organizaci v servisových balíčcích a v subsystémech a dále pak organizaci těchto subsystémů.

Systém je založený na často používané třívrstvé architektuře. Její výhodou je, že každá vrstva poskytuje navenek určité rozhraní, přes které s ní může druhá vrstva komunikovat. Není tedy žádný problém, např. změnit poskytovatele datového úložiště. Patří tam prezentační vrstva, aplikační vrstva a datová vrstva.

Prezentační vrstva obsahuje funkce uživatelského rozhraní, které zajišťuje uživateli jak zobrazování informací, tak zpracování jeho zadaných dat. Může existovat několik prezentačních vrstev pro různé druhy zařízení, platformy a prostředí. Tato vrstva také zodpovídá za validaci údajů zadaných uživatelem.

Prezentační vrstva využívá návrhového vzoru Model-Pohled-Řadič, kde **model** zastupuje datové objekty, které přenášejí data mezi jednotlivými objekty. **Pohled** znamená to, co zajišťuje zobrazení výstupu na uživatelské obrazovce. V případě tohoto systému zajišťuje tuto část GUI aplikace pomocí Swing knihovny J2SE, a nakonec **řadič**, který zastupuje třídy a konfigurační soubory, které zajišťují instalování vhodných objektů a naplnění těchto objektů daty.

Aplikační vrstva tvoří prostředníka mezi vrstvou prezentační a vrstvou datovou. Přebírá data z vrstvy prezentační, provede na nich požadované operace a výsledky těchto operací uloží do datové vrstvy, nebo naopak požádá datovou vrstvu o všechna požadovaná data, ty zpracuje potřebnými operacemi a pak je posle prezentační vrstvě, která tyto finální výsledky zobrazí uživateli.

Datová vrstva obsahuje funkce pro přístup k informacím v datovém úložišti. Tato vrstva je nejnižší vrstvou celého systému. Jejím úkolem je sestavení struktury příkazu pro databázi, kterou doplní o data získaná z aplikační vrstvy, tento příkaz pak při spojení s databází provede.

V systému se využívá spojení s databází pomocí unifikovaného přístupu JDBC. Jeho základem je využití funkčnosti poskytované JDBC ovladačem, který překládá do nativních volání dané databáze. Díky tomu je aplikační vrstva odstíněna od specifického rozhraní databáze a může se naučit jednotné rozhraní JDBC, které pak použije pro přístup do libovolné databáze, která poskytuje JDBC ovladač. V dnešní době to jsou prakticky všechny hlavní systémy, ovladače jsou optimalizované a vyvíjené samotnými výrobci databázových strojů.

Mezi prezentační a aplikační vrstvou existuje autorizační mechanismus, popsáný ve Specifikaci požadavků jako „Přihlášení“. Ten zjišťuje, zda daný uživatel má přístup k funkcím systému.

Podobný systém je i mezi aplikační a datovou vrstvou, kde se obecně zjišťuje, zda má aplikační vrstva potřebné informace k přístupu do databáze.

Pohled procesů

Je popis jednotlivých procesů a vláken v architektuře. Popisuje organizaci procesů do skupin a také poukazuje na akce mezi jednotlivými procesy. Jelikož bude software po první iteraci jen prototypem, bude zde jediný hlavní proces, který tvoří hlavní aplikace.

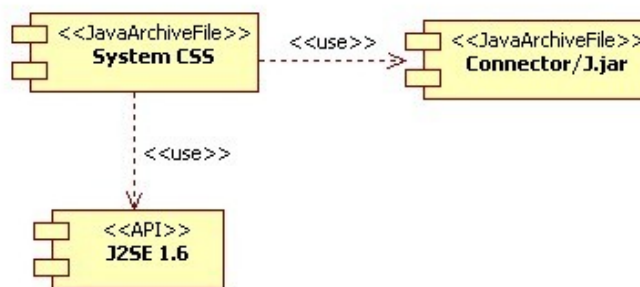
Pohled implementace

Pohled implementace zachycuje soubory a komponenty tvořící výsledný kód systému a jejich vzájemné závislosti. Struktura pro první iteraci je znázorněna na obrázku 5-14.

Komponenta Connector/J je JDBC ovladačem pro přístup k databázovému systému MySQL a bude použita nejnovější osvědčená a otestovaná verze 5.1.10.

Komponenta Java 2 Standard Edition je rozhraní pro programování aplikací (Application Programming Interface) verze 1.6.

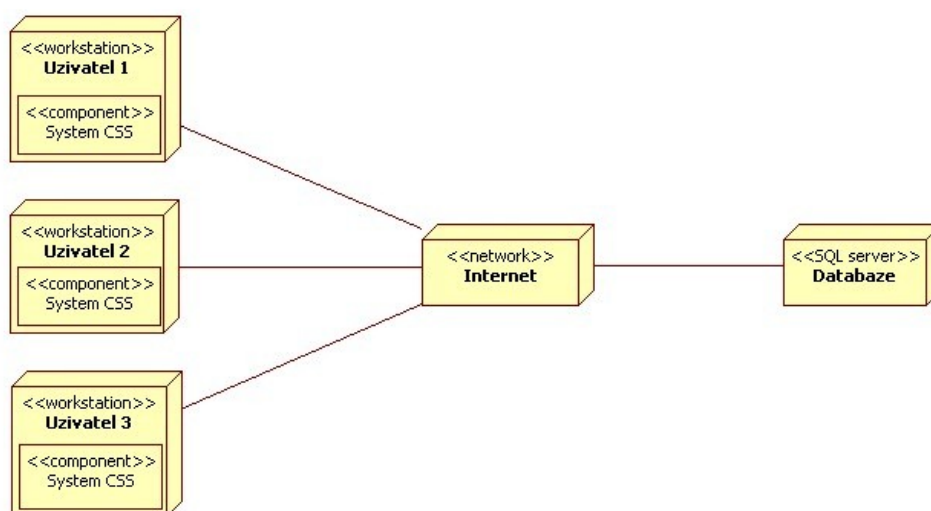
Komponenta Systém ČSS obsahuje všechny funkční třídy projektovaného systému.



Obrázek 5-14 Diagram komponent: Pohled implementace

Pohled nasazení

Pohled nasazení popisuje fyzické konfigurace, na které pak bude software aplikován a jak konkrétně budou jednotlivé komponenty začleněny. Diagram nasazení je zobrazen na obrázku 5-15.



Obrázek 5-15 Diagram nasazení: Pohled nasazení

Workstation jsou jednotlivé pracovní stanice uživatelů, je to obecné označení osobních počítačů nebo notebooků. Součástí je komponenta Systém ČSS, která obsluhuje aplikaci pro potřeby uživatele. Zde také dochází k validaci vstupních dat a zobrazování výsledku operací. Podmínkou je, že jednotlivé stanice jsou napojeny k síti (**network**) Internet, který zde plní funkci přemostění k **SQL serveru**, který obsahuje persistentní data a zpracovává transakce.

Popis základních schopností nového systému

Na základě vyhodnocení specifikací požadavků jsem vyhotovil seznam utilit, které musí systém obsahovat. Jednotlivé schopnosti mají za úkol vylepšení práce se systémem a to rychlostí, přehledností, validací a komunikací s uživatelem.

Validace vstupních dat je mechanismus kontroly zadávaných textových údajů od uživatele. V první iteraci se budeme zabývat kontrolou zadávání povinných údajů, k čemuž stačí deklarace specifických direktiv. Aplikace tedy bude do databáze odesílat všechny potřebné údaje (v našem případě se jedná o všechny odesílané hodnoty).

Indikátor času slouží k příjemnému a rychlému vyhledání aktuálního datumu a času při jakémkoliv průchodu systémem. Jeho zobrazení je umístěno na vždy viditelné části aplikace a při průchodu hlouběji do její struktury.

Zobrazení stromové struktury patří mezi důležitý prvek přehlednosti. Slouží k vyobrazení složitých strukturovaných objektů, které obsahují odkazy na své rodiče. Navíc akce jednotlivých uzlů je možno předefinovat například pro výběr skupiny objektů, spuštění více aktivit najednou apod. V systému této techniky hlavně využijeme k vykreslení struktury adresářů a jejich souborů, což je pro uživatele, zvyklé pracovat s adresářovou strukturou operačního systému Windows, jistě mnohem přehlednější než seznamy.

Zobrazení seznamu je výhodné při zavádění určitých položek, čehož využijeme u evidence členu ZO. Informace o jednotlivci jsou rozsáhlé a neustále je všechny vypisovat by bylo velmi nepřehledné, proto je pro uživatele k nahlédnutí kompletní seznam se základními údaji a po aktivaci editace se objeví další údaje.

Tyto seznamy lze také použít k třídění obsahu dle vybraných atributů nebo k exportování do forem vhodných pro tisk. Další výhodou seznamu je možnost využití filtrování k zamaskování nepotřebných položek a tak dosáhnout maximální přehlednosti.

Horní menu zajišťuje hned několik výhod. První je zachování struktury dle konvencí operačního systému Windows a druhá nabízí rychlý přístup k základním funkcím systému. Tyhle rolovací panely také šetří místo zobrazované plochy aplikace a ta pak zbytečně nezabírá velkou část pracovní plochy.

Záložkový systém se obvykle u aplikací nepoužívá, slouží spíše k organizaci dokumentů. Dokáže však velké množství ploch skladovat na jednom místě a uživatel se mezi nimi přepíná jen podle potřeby. Všechny podsystémy, které jednotlivé záložky spravují, však zůstávají činné, proto je zde výhodná kombinace s horním menu.

Zprávy a hlášky jsou důležitými prvky systému, které vylepšují komunikaci systému s uživatelem. Ten je v průběhu své práce co nejdříve informován o chybě nebo změně, aby na ní mohl reagovat. Taky je potřeba počítat s tím, že uživatel chybí, proto by se systém měl opakovaně dotazovat na provádění rozhodujících úkonů, primárně těch, které mají destruktivní účinky (například mazání dat).

Nápovědy by měl obsahovat každý systém, se kterým pracují běžní uživatelé. Nápověda je však obvykle realizována v takové fázi projektu, kdy už nedochází k zásadním změnám jeho základní funkčnosti.

Moduly prototypu

Moduly jsou osamostatněné části systému, kdy každý v sobě slučuje podobné funkce. Moduly fungují samostatně, ale mívají mezi sebou vazby. V první iteraci se zaměříme na čtyři základní moduly, které vycházejí ze specifikace požadavků. Jsou to přihlašovací modul, modul pro evidenci členů, modul pro správu dat a dokumentů a modul pro plánování událostí.

První modul je globální, obsluhující přístupy ke všem ostatním a je označen jako **přihlašovací modul**. Tento modul obsahuje nejen funkce pro přihlášení uživatele, které je podmíněno správně zadaným přihlašovacím číslem a heslem uživatele, ale také pro odhlášení uživatele. Tato funkce zajišťuje vymazání všech přihlašovacích údajů odhlášením se uživatele a neukončí při tom systém.

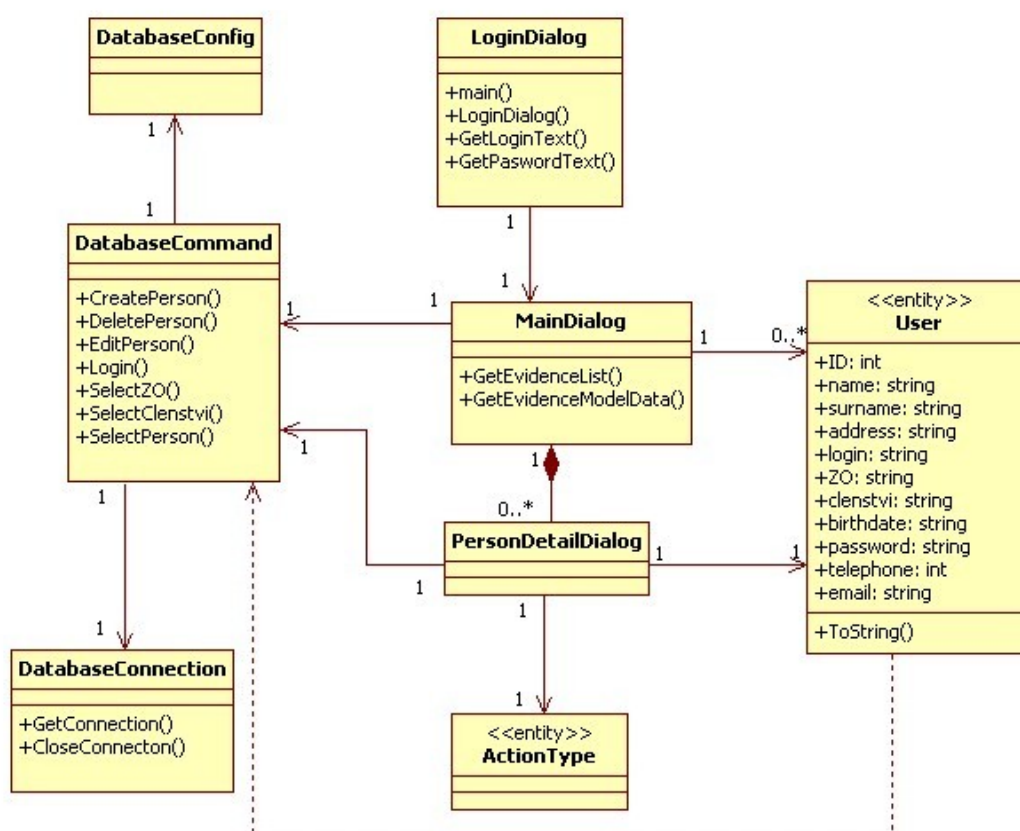
Další částí je **modul pro evidenci členů**, který slouží k záznamu informací o členech ČSS. Obsahuje funkce pro zobrazení seznamu všech členů, kde lze jednoduše zobrazit soupis všech členů a jejich základní informace, dále funkce pro přidávání nových členů či editace členů již zavedených, což umožňuje dynamicky aktualizovat seznam členů. Funkce editace členů zároveň slouží jako přístup k podrobným informacím o vybrané osobě. Poslední funkce mazání členů je nutná pro vyhovění vyhlášce o skladování osobních informací, když po odchodu člena musí být veškeré informace o něm natrvalo odstraněny.

Modul pro správu dat a dokumentů je řešením na otázku přenosu dat mezi jednotlivými členy či uskupeními. Obsahuje funkce pro ukládání, editaci a mazání souborů do stromového systému, čímž zaručuje jednoduchou orientaci v jejich struktuře. Také obsahuje jednoduchou funkci pro vyhledávání souborů dle jejich názvů.

Posledním prvkem je **modul pro plánování událostí**, který zajišťuje přehledné zobrazení veškerých aktivit v ČSS se základním popisem aktivity, termínem, lokací nebo odkazem na osobu, která aktivitu zastřešuje a organizuje. Aktivity je možno přidávat, upravovat nebo mazat, takže uživatelé mají možnost rychlé reakce na aktuální změny.

Popis konkrétního návrhu objektů

Tato podkapitola nabízí základní statický pohled na systém, jeho jednotlivé třídy, jejich obsah a statické vztahy mezi nimi. Statický model vykreslený diagramem tříd je zobrazen na obrázku 5-16.



Obrázek 5-16 Třídní diagram: Statický model systému

LoginDialog je hlavní třída, která inicializuje aplikaci systému a obsluhuje funkce modulu pro přístup do systému.

MainDialog obsahuje základní aktivní okno systému a přístup k jednotlivým modulům systému pomocí panelů.

PersonDetailDialog slouží vyobrazení formuláře pro přidání nového člena a editování stávajícího člena v modulu evidence a také zajišťuje verifikaci vstupních dat.

ActionType je vedlejší třídou k vytvoření nového datového typu dle pro akce v modulu evidence (editování a mazání)

User osahuje funkce pro vygenerování seznamu evidovaných členů a k získávání či ukládání informací o jednotlivých členech.

DatabaseConfig je konfiguračním objektem pro nastavení přístupu k databázi a jejím jednotlivým položkám.

DatabaseCommand slouží k převedení požadavku aplikace do SQL příkazu pro databázi.

DatabaseConnection vytváří nebo ukončuje propojení s databází pomocí JDBC ovladače.

5.2.5 Implementace – 1. iterace

V této disciplíně je nejdůležitější pro zpětnou kontrolu splnění specifikovaných požadavků formou implementování navržené architektury do spustitelného kódu prototypu.

Implementovaný prototyp obsahuje funkce pro přihlášení, odhlášení uživatele a systém evidence členů ČSS. Také je zde implementován náhled na další dvě, ještě nezpracované funkce, a to na systém ukládání, vyhledávání a stahování veškerých dokumentů vytvořených v rámci ČSS a náhled systém evidence akcí a zahraničních expedicí. Jedná se jen o grafické zpracování, aby zadavatelé měli možnost lepšího náhledu na komplexnost sestavovaného systému.

Uživatelské rozhraní je vytvořeno na konvencích operačního systému Windows, což splňuje požadavky zadavatelů.

(Zdrojové kódy prototypu je možno najít v příloze č. 6, vygenerovaná dokumentace prototypu pomocí javadoc je v příloze č. 7.)

5.2.6 Testování – 1. iterace

Cílem testování je vytvořit a provést soubor testů pro ověření funkčnosti komponent systému a jejich správné integrace.

Testování probíhalo průběžně pomocí manuálních testů. Jednotlivé kroky testování a očekávané výsledky nebylo nutné zaznamenávat, protože jsem na projektu pracoval sám a zprávu o testu nebylo nutné nikam dále předávat. Ke zkušebním procesům bylo využito MySQL serveru Apache, který umožní simulování přístupu k databázi přes lokální port.

Testováno bylo přihlášení uživatele, odhlášení uživatele, evidence nového člena, editace stávajícího člena, mazání člena.

U funkce **přihlášení uživatele** se testovaly základní ochranné prostředky systému, zda je ke vstupu nutné správné identifikační číslo a heslo, zda systém nehledí na velikosti písmen při zadávání a jestli zápis hesla je vizuálně skryt.

U funkce **odhlášení uživatele** se testovalo, zda systém odstraní poslední zadávané přihlašovací údaje z paměti i po havárii systému nebo po jeho opětovném spuštění.

U funkce **evidence nového člena** se testoval hlavně zadávací formulář, který musí reagovat na základní chyby vzniklé při zadávání informací a také jestli požadované informace byly skutečně vloženy do databáze.

U funkce **editace stávajícího člena** se prováděly stejné testy jako u evidence nového člena, ale navíc se také testovalo, zda došlo ke správnému načtení informací o členovi z databáze.

U funkce **mazání člena** se testovalo, zda došlo k odstranění zvoleného člena z databáze a zda je vymazání informací nevratné.

Všechny finální testy jednotlivých utilit systému proběhly úspěšně.

5.2.7 Ostatní disciplíny – 1. iterace

Jelikož se tyto disciplíny zabývají buď finální částí projektu, nebo týmové spolupráce, nesoustředil jsem na ně tolik pozornosti a nebylo na nich odvedeno tolik práce, jako na disciplínách předchozích.

Nasazení

Při nasazení se aplikovaly konfigurace hardwarového prostředí dle pohledu nasazení uvedeného v popisu architektury systému.

Správa prostředí

K modelování za pomoci jazyka UML k vytvoření vizuálních obrazů a diagramů bylo použito open-source nástroje StarUML [13]. Textové dokumenty vznikaly v textovém procesoru Microsoft Office Word 2007 [14]. Implementace byla prováděna v open-source vývojovém prostředí Eclipse verze Helios 2 [15]. K implementování grafického rozhraní bylo využito open-source nadstavby pro prostředí Eclipse CloudGarden's Jigloo [16].

5.3 Nový systém – 2. iterace

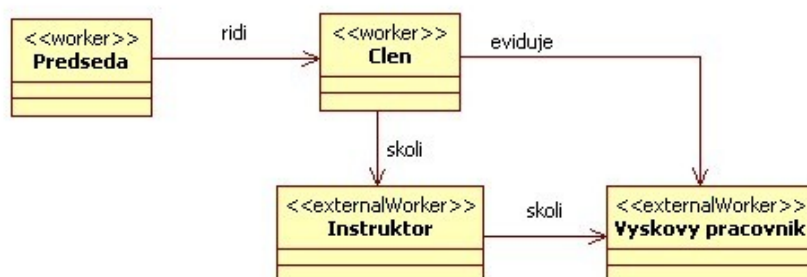
Druhá iterace má za cíl modelovat a analyzovat procesy komise pro práci ve výškách a nad volnou hloubkou a specifikovat požadavky na základě těchto procesů. Ve zbývajících disciplínách popíši jen základní změny a vylepšení od první iterace.

5.3.1 Byznys modelování – 2. iterace

Na základě priority procesů jednotlivých uskupení se budeme v druhé iteraci zabývat základními procesy komise pro práci ve výškách a nad volnou hloubkou, jejíž obecná definice je již v kapitole 3. Česká speleologická společnost. Pro další průzkum problematiky jejich procesů je opět nutné popsat její strukturu až na základní prvky. Tyto informace byly stejně jako u procesů ZO získány především z dotazníků a briefingu. Tyto dokumenty je možno najít v příloze č. 2.

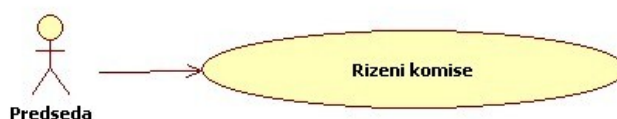
Organizační struktura komise pro práci ve výškách a nad volnou hloubkou

Všechny komise v ČSS se obecně skládají z členu komise a jejího předsedy, jejich procesy se ale také týkají externích účastníků (instruktorů technologie práce ve výškách a nad volnou hloubkou a výškových pracovníků), kdy cílem komise je právě uspokojení jejich potřeb a zároveň evidence těchto osob, jak je znázorněno na obrázku 5-17.



Obrázek 5-17 Organizační struktura komise pro práci ve výškách a nad volnou hloubkou

Předseda komise distribuuje práci mezi jednotlivé členy a řídí tak průběh jednotlivých aktivit, které rozloží na menší bloky a přiřadí dle odborností a možností členů. Jeho základní procesy jsou znázorněny následujícím diagramem na obrázku 5-18.



Obrázek 5-18 Základní procesy předsedy komise

Člen komise je výkonný prvek tohoto uskupení. Jeho hlavními úkoly je evidence výškových pracovníků a instruktorů pro vlastní přehled a také pro splnění vyhlášky o školení pracovníků. Také slouží jako recenzenti nových nebo stávajících pracovních pomůcek a analyzují a propagují úpravy v legislativě oborů výškových prací a prací nad volnou hloubkou, jak je znázorněné na obrázku 5-19.



Obrázek 5-19 Základní procesy člena komise

Instruktor a výškový pracovník z pohledu procesů komise žádné aktivity neinicializují, ale jsou jejich součástí. Instruktor je jakýmsi přemostěním znalostí a pravomocí mezi členy komise a výškovými pracovníky. Mají možnost školit osoby, pro které je výstupní certifikát v jejich pracovních procesech nutností, ale nemohou školit nové instruktory. Výškový pracovník je nejnižší stupeň této struktury s možností tyto speciální aktivity vykonávat.

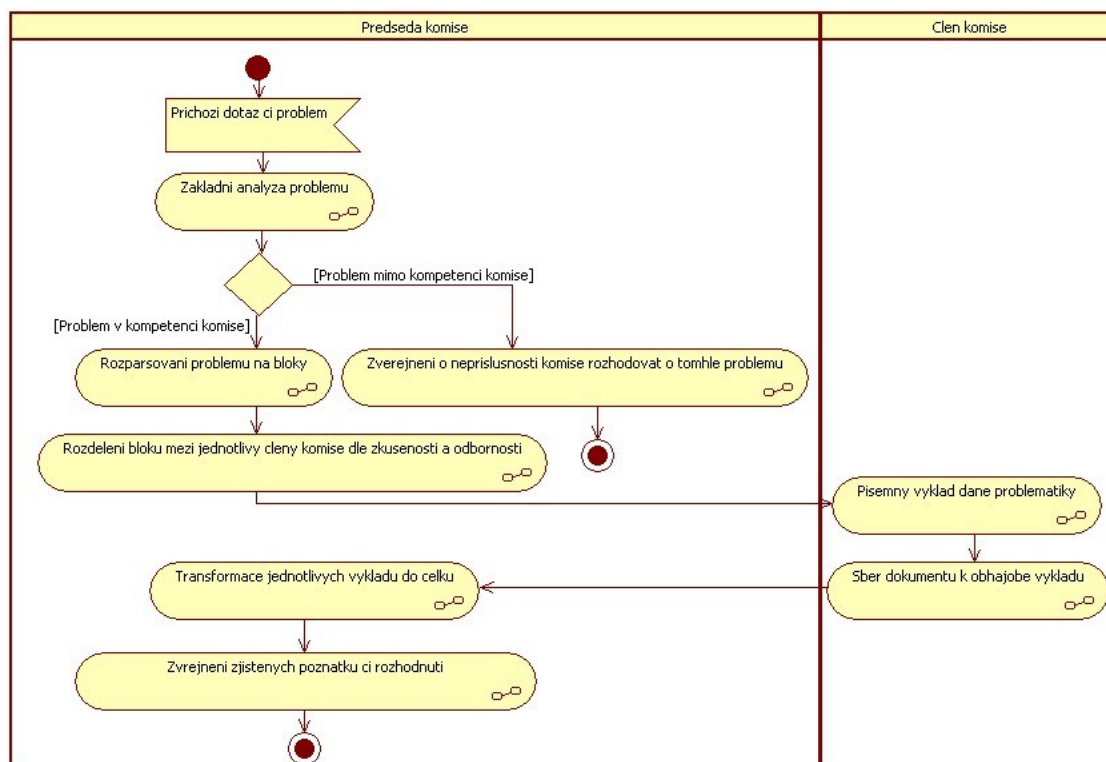
Specifikace hlavních procesů komise

V tomto odstavci se zaměříme na modelování nejdůležitějších procesů komise pro výškové práce a práce nad volnou hloubkou, které musí být provedeny pro úspěšné splnění požadovaného úkolu. Popis procesů tak umožňuje řídit a monitorovat jednotlivé prováděné činnosti. V případě potřeby můžeme procesy optimalizovat či navrhnout úpravy pro efektivnější využití zdrojů. Ke každé specifikaci je přiložen přehledový diagram toku činností daného procesu.

Mezi základní procesy patří řízení komise, evidence vyškolených pracovníků a instruktorů a školení instruktorů.

Řízení komise je v kompetenci předsedy komise. Jeho hlavní náplní je analýza dotazů, dokumentů, problémů či připomínek kladených na komisi pro práci ve výškách a nad volnou hloubkou. Rozhoduje o tom, zda je komise kompetentní podat odborné vyjádření. Pokud ano, pak rozdělí analýzu mezi jednotlivé členy na základě jejich znalostí a profesní odbornosti. Členové jsou povinni ke svým vyjádřením zajistit odbornou literaturu, která potvrdí jejich stanovisko (nejčastěji se jedná o citaci norem a zákonů). Výsledek této práce pak předseda komise prezentuje veřejnosti nebo přímo osobě, která dala podnět k této práci.

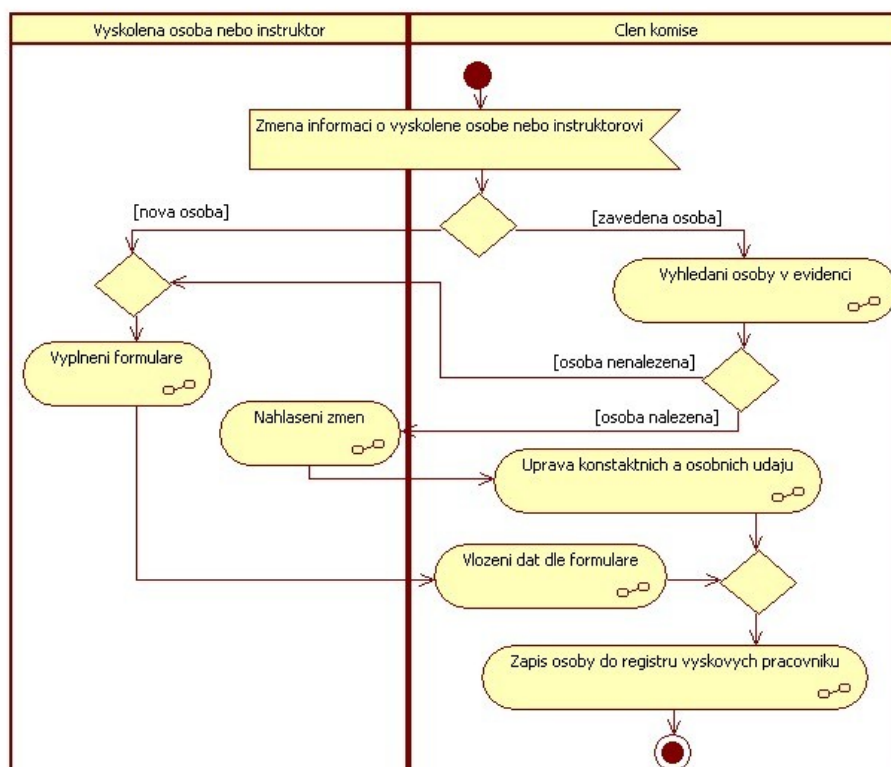
Diagram toku činností procesu (viz obrázek 5-20).



Obrázek 5-20 Diagram aktivit: Řízení komise

Evidence vyškolených pracovníků a instruktorů obstarávají členové komise. Jedná se o vypracování dokumentu, který je seznam všech osob, které byly školeny pomocí speleo techniky. Nutné je zaznamenávat jak nové osoby, tak upravovat ty zavedené, aby byly informace aktuální. Důležitým pravidlem je zachování i již neaktivních pracovníků po dobu minimálně deseti let, jelikož se komise zavazuje, že osoba byla vyškolená dle předpisů a v případě úrazu jsou tak právně krytí. Informace o osobách se získávají buď ve formě přijímacích formulářů, nebo při úpravě dat změny nahlásí samotní pracovníci, jelikož nesprávné údaje činí jejich certifikát neplatný.

Diagram toku činností procesu (viz obrázek 5-21).

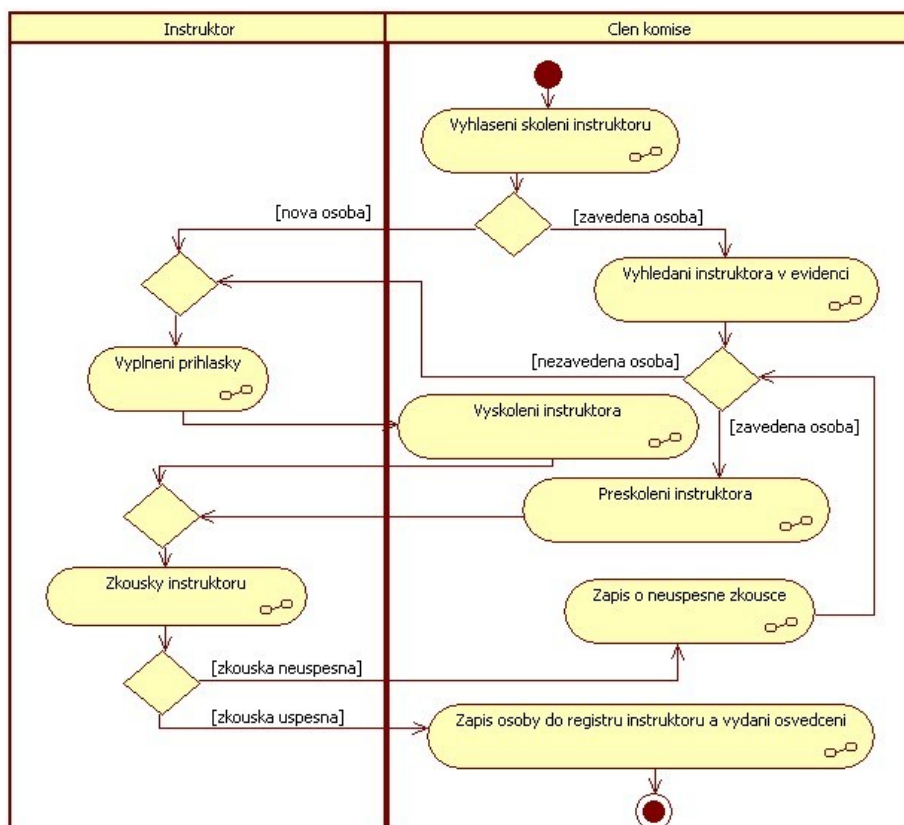


Obrázek 5-21 Diagram aktivit: Evidence vyškolených pracovníků a instruktorů

Školení instruktorů je aktivita nutná k akreditaci osob, které mohou dále vyučovat, přeškolovat a certifikovat nové výškové pracovníky. Podmínkou je alespoň jednou projít teoretickou a praktickou výukou a minimálně jednou za dva roky zdárně složit písemné zkoušky ve formě testu (tyto testy se každý rok obměňují). Předpokladem jsou základní znalosti speleo technik. Neúspěšné zkoušky se evidují, ale obvykle nejsou brány v potaz při nástupu k další zkoušce (jen ve výjimečných případech mnohonásobného neúspěšného opakování).

Diagram toku činností procesu (viz obrázek 5-22).

Podrobnější specifikaci těchto procesů a obecnou specifikaci vedlejších procesů, jako shromažďování informací o pomůckách a sledování vývoje legislativy je možno najít v příloze č. 8.



Obrázek 5-22 Diagram aktiv: Školení instruktorů

Návrhy na vylepšení těchto procesů

Při studování a modelování jednotlivých procesů jsem narazil na některé nesrovnalosti nebo nevhodně volené postupy. Většina z nich je způsobená neexistujícím systémem této společnosti.

Evidence instruktorů a výškových pracovníků

Jelikož pracovníky školí jednotliví instruktoři, je zde podobný problém jako u evidence členů ČSS. Seznamy se velmi složitě přemísťují či skládají a také je velmi složité vyhledávat změny, pokud na ně není přímo upozorněno. Tím vznikají časté chyby, které se nechvalně promítají i mimo společnost, když o informace, zda dotyčná osoba byla opravdu proškolená v daném časovém období, zažádají jiné společnosti, firmy nebo právní složky.

Systém bude obsahovat ucelenou evidenci všech instruktorů a výškových pracovníků, kdy členové komise pro práci ve výškách a nad volnou hloubkou budou mít k jednotlivým záznamům přístup a také budou moci jednoduše vybraný záznam exportovat a tak předat dál. Tato evidence bude uložena v databázi, takže šíření změn bude téměř okamžité.

Správa řešených problému, dokumentů a připomínek

Veškeré řízení těchto aktivit probíhá na sezeních komise, které ale nebývají příliš často, jelikož jsou členové komise z různých míst České republiky. Tento proces přitom nevyžaduje, aby členové komisi mohli komunikovat v reálném čase, bloky jim přidělí předseda (někdy i na základě domluvy) a

pak už na nich pracují samostatně. Předseda navíc úkoly přiděluje až na následujícím sezení a dalším následujícím sesbírává vyjádření.

Distribuci dokumentů a práce na nich jako i pak jejich sesbírání bude obstarávat systém, kde jak členové, tak předseda mohou vidět aktuální stav práce. Tím se eliminuje nutný interval mezi jednotlivými sezeními komise. Systém také zaručí jednoduchý přehled o práci vykonané a té, která je třeba ještě udělat.

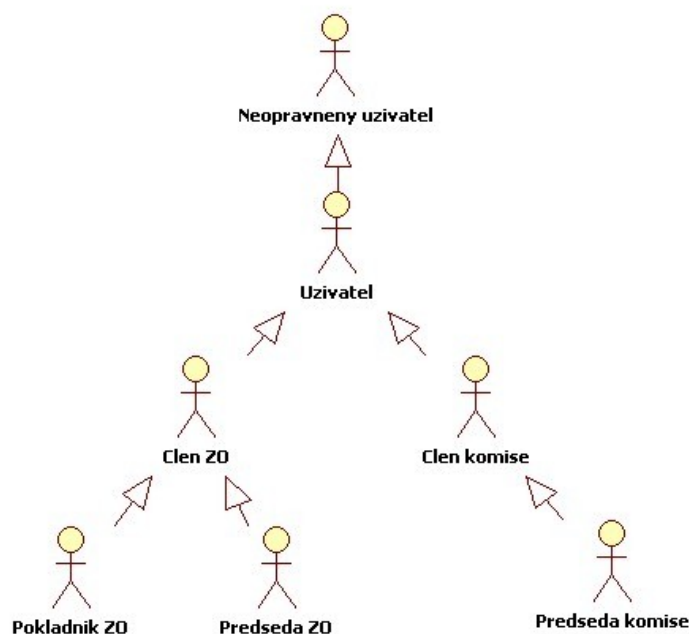
Přezkoušení instruktorů

Tato aktivita se provádí vždy několikrát za rok, kdy je nutné, aby již zavedené osoby přijeli a vypsalí potřebné testy (u každé osoby však minimálně jednou za dva roky, ale neexistuje všem vyhovující termín). Testy se pak uchovávají jako písemný doklad o splnění jedné z podmínek pro získání akreditace ke školení výškových pracovníků.

Prakticky se jedná o akt vyplnění testů a vyhláška nestanoví, kde dotyčný musí testy provádět, ani jestli je zapotřebí dozor, který při přezkoušení nebývá. Tyto testy by tedy bylo možné provádět přímo pomocí systému a tím by se eliminovala nutná cesta do města, kde se právě školení instruktorů provádí. Systém by taky automaticky vyhodnotil výsledky a tím by ušetřil čas zkoušejícím členům komise.

5.3.2 Specifikace požadavků – 2. iterace

V druhé iteraci se zaměříme na podrobnější specifikaci rolí v systému a na zabudovanou evidenci instruktorů a výškových pracovníků. Role oprávněného uživatele (obecně ho nadále budeme nazývat uživatelem) bude podrobně rozčleněna a stejná zůstane jen role neoprávněného uživatele. Strukturu rolí zobrazuje obrázek 5-23.

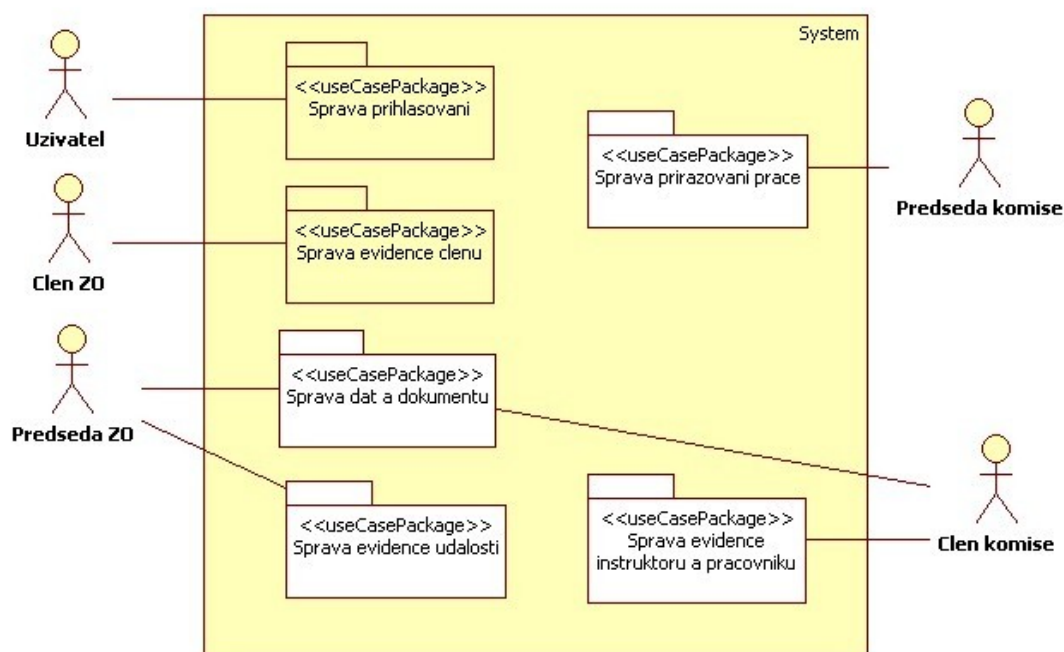


Obrázek 5-23 Struktura rolí - 2. Iterace

Role **Uživatel** zde zapouzdřuje všechny osoby, které mají přístup do systému, a to znamená, že jejich registrační údaje jsou již v systému zavedeny. Tato role se pak dělí na dvě skupiny. Je to **Člen ZO**, který zastupuje všechny osoby spadající pod uskupení ZO a **Člen komise**, který obsahuje všechny osoby patřící pod komisi pro práci ve výškách a nad volnou hloubkou. **Pokladník ZO** (má přístup k nastavení finančního plánování událostí) a **Předseda ZO** (má přístup ke správě evidence členů ZO, ke správě událostí a aktivit a také má možnost upravovat nebo odstraňovat dokumenty z částí pro správu dat a dokumentů) jsou pak speciální role z balíku členů ZO, které mají přístup ke svým specifickým funkcím. **Předseda komise** je speciálním případem skupiny členů komise, který má přístup k funkcím pro řízení komise.

Funkční požadavky

Nyní můžeme na základě rolí přidat nové požadavky kladené na systém. Jednotlivé funkce budou definovány jako případy užití, jak je zobrazeno na obrázku 5-24. Nové funkční požadavky jsou vyznačeny bílou výplní.



Obrázek 5-24 Diagram případů užití: Funkční specifikace požadavků - 2. iterace

Přiřazování práce bude obsahovat funkce pro přidělení bloků jednotlivých problémů, připomínek a dotazů směřovaných na komisi pro výškové práce a práci nad volnou hloubkou. Tyto bloky rozdělí předseda komise a pomocí správy dat a dokumentů budou přístupné jednotlivým členům komise.

Správa dat a dokumentů je řešením na otázku přenosu dat mezi jednotlivými členy či uskupeními. Obsahuje funkce pro ukládání, editaci a mazání souborů do stromového systému, čímž zaručuje jednoduchou orientaci v jejich struktuře. Také obsahuje jednoduchou funkci pro vyhledávání souborů dle jejich názvů.

Evidence událostí zajišťuje přehledné zobrazení veškerých aktivit v ČSS se základním popisem aktivity, termínem, lokací nebo odkazem na osobu, která aktivitu zastřešuje a organizuje. Aktivity je možno přidávat, upravovat nebo mazat, takže uživatelé mají možnost rychlé reakce na aktuální změny.

Evidence instruktorů a pracovníků slouží k záznamu informací o vyškolených osobách v rámci komise pro výškové práce a práce nad volnou hloubkou. Obsahuje funkce pro zobrazení seznamu všech pracovníků a instruktorů, kde lze jednoduše zobrazit jejich soupis a jejich informace o nich, dále funkce pro přidávání nových osob či editace osob již zavedených, což umožňuje dynamicky aktualizovat seznam členů. Funkce editace osob zároveň slouží jako přístup k podrobným informacím.

5.3.3 Analýza a návrh – 2. iterace

V druhé iteraci se část analýza zaměřuje na nově vzniklé objekty a část návrhu slouží k popisu nově vzniklých modulů.

Nové třídy objektů

V druhé iteraci došlo k rapidní změně zavedením struktury rolí a také vnořením nových specifikací, což se projevuje i vznikem nových objektů. Vizualní modelování jednotlivých částí je znázorněno v příloze č. 9.

K balíčku **Uživatel**, který obsahuje všechny možné informace a funkce pro vedení evidence uživatelů je nutné přidat **Výškový pracovník**, který bude obsahovat informace nutné k evidenci instruktorů a výškových pracovníků, jako datum platnosti jejich certifikátu, informace o provedených zkouškách a absolvovaných školeních.

Pro potřeby autorizace je zaveden vnořený balíček **Přístupová práva**, který obsahuje informace o uživateli a jejich privilegiích v rámci modulů systému.

Objekt **Dokumenty** obsahuje veškeré informace o vložených dokumentech a souborech, aby bylo možné v seznamu jednoduše vyhledat požadovanou položku. **Správa dokumentů** pak obsluhuje funkce pro vyhledávání, specifikaci, ukládání nebo mazání těchto položek. Je zde také zavedeny balíček zastupující seznam jednotlivých nadefinovaných **Typů dokumentu**, aby bylo možné identifikovat a sortovat soubory nejen podle názvu.

Balíček **Události** zahrnuje data o jednotlivých akcích a událostech včetně osob, které událost založili či se na ní přihlásili. **Správa událostí** obstarává funkce pro zakládání, úpravu nebo zrušení těchto aktivit a také zajišťuje možnost přihlášení se k této události. Budou zde vnořené objekty **Lokalita**, **Vybavení** a **Datum**, které slouží k bližšímu popisu události, k nadefinování počátku a přibližného konce akce a k popisu nutného vlastního vybavení (popř. vybavení, které zajistí pořadatel).

Posledními novými objekty jsou balíček **Práce**, který obsahuje informace o dotazech, připomínkách či žádostech směřovaných na komisi a balíček **Řízení práce**, který slouží přidělení těchto položek mezi vybrané členy komise. Balíček **Typ práce** obsahuje předdefinovaný seznam, dle kterého je možné jednoduše třídit tyto aktivity. Pro kontrolu a zajištění plnění plánu je zavedena část **Termín**, která obsahuje datum, jak zahájení, tak odhadovaného zakončení celé práce, nebo jednotlivých částí, které budou zastupovat balíček **Blok**. Je nutné evidovat, co bylo komu přiřazeno, v jaké fázi se práce nachází.

Nové moduly prototypu

V druhé iteraci byly přidány dva nové moduly, které vycházejí ze specifikace požadavků. Je to modul přiřazování práce a modul pro evidenci instruktorů a výškových pracovníků.

První nový **modul pro přiřazování práce** slouží k vkládání nových požadavků na komisi pro práci ve výškách a nad volnou hloubkou předsedu komise, který rozdělí jednotlivé úkony mezi členy komise. Ti pak k nim budou mít přístup pomocí modulu pro správu dat a dokumentů. Tento modul je určený jen pro předsedu komise a jiný uživatel k němu nemá přístup.

Další částí je **modul pro evidenci instruktorů a výškových pracovníků**, který slouží k centralizaci informací o jednotlivých osobách s možností provádění změn u osobních informací, nebo k přidávání osob nových. Zde je důležitá část pro aktivaci osob, které zdárně splnily podmínky školení a byl jim tak prodloužen certifikát. Neexistuje zde funkce pro mazání, protože je nutné tyto informace zálohovat deset let a pro komisi je výhodnější, když budou mít přístup k těmto informacím navždy.

5.3.4 Ostatní disciplíny – 2. iterace

Ostatním disciplínám druhé iterace už nebylo věnováno příliš pozornosti, protože jsou nad rámec požadavků na tuto diplomovou práci. Vývoj systému však nebyl zastaven a bude se v něm i nadále pokračovat.

Implementace prototypu bude zahrnovat všechny definované moduly z první i druhé iterace a také bude přidán balíček pro nastavení a rozpoznávání přístupových práv jednotlivých členů ČSS.

Testování bude stejně jako v první iteraci prováděno průběžně a to manuálním testem. Funkcionální a performance testy budou prováděny až ve fázi Testování.

Jelikož je patrné, že ve vývoji toho systému bude nutné řady iterací, bude v disciplíně **Řízení změn a konfigurací** zaveden systém pro správu verzí CVS [17]. Ten zajistí záznam historie, zápis, identifikaci a slučování oprav jak dokumentace, tak částí implementace.

6 Závěr

Cílem této diplomové práce bylo analyzovat základní procesy vykonávané Českou speleologickou společností a na základě takto popsanych procesů měla být provedena analýza a návrh systému, který umožní evidenci a správu jednotlivých prvků procesu.

Pomocí iterativní metody dle metodiky RUP jsou detailně popsány procesy dvou nejvýznamnějších uskupení této organizace (Základní organizace a Komise pro práci ve výškách a nad volnou hloubkou) a v rámci této analýzy jsou přiloženy návrhy na jejich vylepšení, které směřují k tvorbě nového systému. Součástí je také plnohodnotná specifikace funkčních i nefunkčních požadavků plynoucích z potřeb těchto uskupení. Vývoj výsledného prototypu byl zahájen analýzou a návrhem pro vytvoření analytického modelu, aby implementace tohoto systému byla co nejsnadnější a zároveň splněny vyžádané standardy.

Nápadů na ulehčení činností Základní organizace a Komise pro práci ve výškách a nad volnou hloubkou je mnoho, protože se jedná o rozsáhlé části České speleologické organizace, ale jako příklad můžeme zmínit vytvoření komplexního výukového a zkušebního systému jak pro instruktory, tak pro výškové pracovníky nebo interaktivní systém pro vkládání připomínek a požadavků, který by ulehčil přiřazení položek členům komise a zamezil přijímání požadavků nespádajících pod kompetence komise. Pro Základní organizace systém pro automatické generování map dle vektorové polygonové sítě nebo využívání kapesních počítačů při sčítání letounů.

Dle specifikace RUP je projekt stále ve fázi Rozpracování. Dalším krokem je tedy upravit prototyp o nové části, analyzovat procesy dalších organizačních jednotek dle priority jejich procesů a nabalování jednotlivých modulů systému k jeho kostře. Rozpracované modely dalších procesů je možno najít v příloze č. 13.

Největším přínosem této práce je v prolomení starých zavedených praktik a prezentace škály nových technologií, které České speleologické společnosti zajistí kvalitní přehled o členech a aktivitách svých složek, a které vytvoří interní systém komunikace navrhnutý speciálně pro potřeby této společnosti.

7 Reference

- [1] Česká speleologická společnost [online]
URL: <http://www.speleo.cz/> [cit. 2011.4.17]
- [2] OMG - Unified Modeling Language [online]
URL: <http://www.uml.org/> [cit. 2011.4.17]
- [3] IBM Rational Unified Process (RUP) [online]
URL: <http://www-01.ibm.com/software/awdtools/rup/> [cit. 2011.4.17]
- [4] Object Management Group [online]
URL: <http://www.omg.org/> [cit. 2011.4.17]
- [5] UML – skladba UML [online]
URL: <http://mpavus.wz.cz/uml/uml-skladba-2.php> [cit. 2011.4.17]
- [6] The Rational Objectory Process [online]
URL: http://www.iscn.com/select_newspaper/object/rational.html [cit. 2011.4.17]
- [7] Rational software [online]
URL: <http://www-01.ibm.com/software/rational/> [cit. 2011.4.17]
- [8] Pseudokras na Liberecku - P. Bosák: Definice pseudokrasu [online]
URL: <http://www.volny.cz/daniel.horacek/PseudoKr.htm> [cit. 2011.4.17]
- [9] Speleoterapie - definice [online]
URL: <http://www.detskaspseleoterapie.cz/speleoterapie.htm> [cit. 2011.4.17]
- [10] Josef Svoboda a spol.: Encyklopedický slovník geologických věd, 1. vyd. Praha: Academia 1983.
- [11] Chiroptera - Letouni [online]
URL: <http://www.savci.upol.cz/letouni.htm> [cit. 2011.4.17]
- [12] Laserové dálkoměry - Disto Leica [online]
URL: http://obchod.geometra-opava.com/html/las_dalk.htm [cit. 2011.4.17]
- [13] StarUML – the opensource UML/MDA platform [online]
URL: <http://staruml.sourceforge.net/en/> [cit. 2011.4.17]
- [14] Microsoft Word – Microsoft Office [online]
URL: <http://office.microsoft.com/cs-cz/word/> [cit. 2011.4.17]
- [15] The Eclipse – an open source development platform [online]
URL: <http://www.eclipse.org/> [cit. 2011.4.17]
- [16] Cloud Garden – Jigloo GUI builder [online]
URL: <http://www.cloudgarden.com/jigloo/> [cit. 2011.4.17]
- [17] CVS – open source version control [online]
URL: <http://www.nongnu.org/cvs/> [cit. 2011.4.17]
- [18] Jozef Jakál a kol.: Praktická speleológia, Martin, Osveta 1982.

8 Přílohy

Přílohy jsou k nalezení na přiloženém kompaktním disku ve strukturovaných adresářích.

1. Dokumentace struktury ČSS
 \CSS\
2. Textové záznamy a dotazníky z briefingů
 \CSS\Dotazníky\
3. Specifikace procesů – 1. iterace
 \1. iterace\
4. Model specifikace požadavků – 1. iterace
 \1. iterace\
5. Model analýzy a návrhu – 1. iterace
 \1. iterace\
6. Zdrojové kódy prototypového systému
 \Prototyp\Zdrojovy kod\
7. Automaticky vygenerovaná dokumentace prototypu Javadoc
 \Prototyp\Javadoc\
8. Specifikace procesů – 2. iterace
 \2. iterace\
9. Model analýzy – 2. iterace
 \2. iterace\
10. Zdrojový soubor kompletního projektu open source nástroje StarUML
 \StarUML\
11. Instalační manuál prototypu
 \Prototyp\
12. Spustitelný prototypový systém
 \Prototyp\
13. Rozpracované modely procesů
 \DalsiIterace\
14. Textový soubor obsahující testovací databázi ve formě SQL skriptu
 \Prototyp\